دانشکده علوم و فنون نوین

گروه بین رشته‌ای فناوری (بخش علوم و فناوری شبکه)

استفاده از یادگیری عمیق برای بازشناسی گفتار فارسی

نام دانشجو:

آرمیتا حجی‌مانی

استاد راهنما:

دکتر هادی ویسی

پایان‌نامه برای دریافت درجه کارشناسی ارشد

در رشته علوم تصمیم و مهندسی دانش

اسفند ۱۳۹۵


بسم الله الرحمن الرحيم

چکیده

به فرآیند تبدیل سیگنال صوتی به متن معادل آن تشخیص گفتار گفته می‌شود. امروزه از روش‌های مختلفی جهت بازشناسی گفتار استفاده می‌شود که مهمترین آن‌ها روش آماری مدل مخفی مارکوف و شبکه عصبی می‌باشد. . یکی از مشکلاتی که هنوز در این حوزه مطرح است، بحث افزایش دقت و کارایی این سیستم‌ها می‌باشد و با توجه به این‌که یکی از راه‌های افزایش دقت سیستم‌های بازشناسی گفتار، بهبود مدل آوایی می‌باشد، در این پایان‌نامه برای اولین بار از شبکه عصبی عمیق حافظه کوتاه مدت ماندگار (LSTM) یک‌طرفه و دوطرفه با لایه خروجی طبقه‌بند زمانی پیوندگرا (CTC) جهت ساخت مدل آوایی فارسی استفاده شده است. از آنجایی‌که سیگنال صوت نمونه‌ای از داده‌های متوالی می‌باشد که در آن‌ها مقدار داده فعلی به داده‌های قبلی وابسته است، شبکه‌های عصبی بازگشتی به دلیل دارا بودن حافظه برای این نوع داده‌ها مناسب می‌باشند. شبکه عصبی حافظه کوتاه مدت ماندگار یک شبکه عصبی بازگشتی است که در آن با جایگزین کردن نرون‌های لایه پنهان با بلوک‌های حافظه، مشکل فراموشی داده‌ها در دنباله‌های طولانی رفع شده است و کارایی بالای خود را در مدل‌سازی داده‌های ترتیبی در کاربردهای مختلف نشان داده است.

همچنین در این پایان‌نامه، از شبکه باور عمیق جهت استخراج ویژگی استفاده شده است و نتایج به‌دست آمده با روش پایه استخراج ویژگی که همان ضرایب کپسترال در مقیاس مل (MFCC) است، مقایسه گردیده است. نتایج به‌دست آمده نشان می‌دهد که استفاده از شبکه عمیق در مقایسه با شبکه یک‌لایه کارایی را بالاتر می‌برد. به‌علاوه، استفاده از شبکه دوطرفه موجب افزایش دقت شبکه در مقایسه با شبکه یک‌طرفه، هم در حالت عمیق و هم در حالت غیرعمیق می‌گردد. نتایج به‌دست آمده با مدل مخفی مارکوف (HMM) مقایسه شده است که نشان می‌دهد، استفاده از شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه با ویژگی‌های حاصل از شبکه باور عمیق در بهترین حالت موجب بهبود دقت تشخیص واج فارسی به‌میزان ۸٫۱٪ در مقایسه با مدل مخفی مارکوف روی مجموعه داده‌های فارس‌دات شده است.

کلمات کلیدی: بازشناسی گفتار فارسی، شبکه عصبی حافظه کوتاه مدت ماندگار، شبکه عصبی بازگشتی، شبکه عصبی عمیق، شبکه عصبی دوطرفه، طبقه‌بند زمانی پیوندگرا.

فهرست مطالب

فهرست شکل‌ها

فهرست جداول

فصل اول: مقدمه و معرفی

۱-۱- مقدمه

به فرآیند تبدیل سیگنال صوتی به متن معادل آن تشخیص گفتار[۱] (ASR) گفته می‌شود. تشخیص گفتار کاربردهای مختلفی دارد که از جمله آن می‌توان به سیستم تایپ، تشخیص فرامین و دستورات صوتی، سیستم‌های اطلاع‌رسانی و همچنین ترجمه گفتار به گفتار اشاره کرد. برای طراحی سیستم‌های تشخیص گفتار روش‌های متفاوتی وجود دارد که از جمله مهم‌ترین آن‌ها، روش آماری مدل مخفی مارکوف[۲] (HMM) [۱, ۲] و روش‌های مبتنی بر یادگیری ماشین[۳] مانند شبکه‌های عصبی[۴] (ANN) [۳, ۴] می‌باشد. یکی از مشکلاتی که هنوز در این حوزه مطرح است، بحث افزایش دقت و کارایی این سیستم‌ها می‌باشد که امروزه شرکت‌ها و دانشگاه‌های مختلفی در سراسر جهان روی آن متمرکز هستند. یکی از ایده‌ها به‌منظور افزایش کارایی و دقت سیستم‌های بازشناسی گفتار که در چند سال اخیر مطرح شده است استفاده از یادگیری عمیق[۵] [۱] می‌باشد. استفاده از یادگیری عمیق در بازشناسی گفتار زبان انگیسی نتایج خوبی را به‌همراه داشته است [۵, ۶] و در زبان فارسی نیز چندین کار در زمینه بازشناسی گفتار با استفاده از یادگیری عمیق صورت گرفته است [۷,

[1] Automatic Speech Recognition (ASR)
[2] Hidden Markov Model (HMM)
[3] Machine Learning
[4] Artificial Neural Network (ANN)
[5] Deep Learning

۸]. از آنجایی‌که افزایش دقت در سطح تشخیص واج[۱] منجر به بهبود دقت کل سیستم می‌گردد، در این پایان‌نامه بر روی افزایش دقت واج سیستم‌های بازشناسی گفتار فارسی با رویکرد یادگیری عمیق تمرکز شده است.

با توجه به این‌که پس از ارائه شبکه عصبی حافظه کوتاه مدت ماندگار[۲] (LSTM) [۹, ۱۰] و به‌دنبال آن حل مشکل فراموشی[۳] در شبکه‌های عصبی بازگشتی[۴] (RNN) از این شبکه به‌صورت گسترده در بازشناسی گفتار انگلیسی استفاده گردیده است [۵, ۶, ۱۱–۱۳]، در این پایان‌نامه به‌منظور افزایش دقت تشخیص واج‌های زبان فارسی با استفاده از شبکه عصبی حافظه کوتاه مدت ماندگار یک‌طرفه[۵]، شبکه عصبی حافظه کوتاه مدت ماندگار دوطرفه[۶] (BLSTM)، شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه[۷] (DLSTM) و همچنین شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه[۸] (DBLSTM) مدل آوایی[۹] فارسی ساخته شده است.

۱–۲– گام‌های طراحی سیستم‌های بازشناسی گفتار

فرآیند طراحی سیستم‌های بازشناسی گفتار شامل دو مرحله آموزش[۱۰] و آزمون[۱۱] می‌باشد. در مرحله آموزش، دو مدل زبانی[۱۲] و آوایی ساخته می‌شود و در مرحله آزمون با استفاده از این دو مدل و واژگان[۱۳] طی فرآیند رمز گشایی[۱۴] دنباله کلمات استخراج می‌گردد. شکل ۱–۱ ساختار کلی سیستم‌های بازشناسی گفتار را نمایش می‌دهد. در ادامه هر یک از این دو مرحله را بررسی می‌کنیم.

---

۱ Phoneme

۲ Long Short Term Memory (LSTM)

۳ Vanishing Gradient Problem

۴ Recurrent Neural Networks (RNN)

۵ Unidirectional Long Short Term Memory

۶ Bidirectional Long Short Term Memory (BLSTM)

۷ Deep Long Short Term Memory (DLSTM)

۸ Deep Bidirectional Long Short Term Memory (DBLSTM)

۹ Acoustic Model

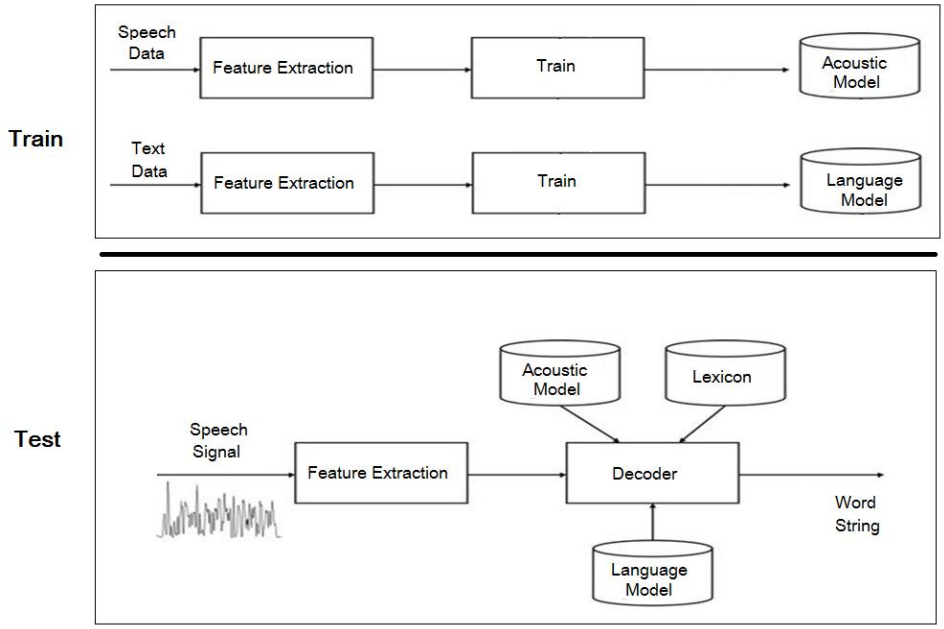
۱۰ Train

۱۱ Test

۱۲ Language Model

۱۳ Lexicon

۱۴ Decoding



شکل ۱-۱) ساختار کلی سیستم بازشناسی گفتار

۱-۲-۱- مرحله آموزش

مرحله آموزش شامل ساخت مدل آوایی و مدل زبانی می‌باشد. مجموعه داده‌های مورد استفاده جهت ساخت مدل آوایی پیکره صوتی[۱] نام دارد که شامل مجموعه‌ای از سیگنال‌های صوتی می‌باشد. همچنین مجموعه داده‌های مورد استفاده جهت ساخت مدل زبانی پیکره متنی[۲] نام دارد و شامل مجموعه‌ای از فایل‌های متنی می‌باشد.

به‌منظور ساخت مدل آوایی در گام اول از سیگنال‌های مجموعه آموزش ویژگی استخراج می‌گردد. در گام بعدی با استفاده از روش‌های آماری مانند مدل مخفی مارکوف یا روش‌های مبتنی بر یادگیری ماشین مانند شبکه‌های عصبی پارامترهای مدل تعیین می‌گردد (مرحله آموزش) و مدل آوایی به‌دست می‌آید. همان‌طور که پیش‌تر نیز گفته شد این پایان‌نامه روی ساخت مدل آوایی فارسی با استفاده از شبکه‌های عصبی حافظه کوتاه مدت ماندگار یک‌طرفه، حافظه کوتاه مدت ماندگار یک‌طرفه و همچنین شبکه‌های عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه و دوطرفه تمرکز دارد.

[۱] Voice Corpus

[۲] Text Corpus

برای ساخت مدل زبانی در گام نخست از داده‌های آموزش که در واقع همان پیکره متنی می‌باشند ویژگی استخراج می‌گردد. سپس با استفاده از روش‌های آماری پارامترهای مدل زبانی استخراج می‌گردد که یکی از معروف‌ترین مدل‌های زبانی N-Gram می‌باشد [۱۴].

۱-۲-۲- مرحله آزمون

در این مرحله ابتدا از سیگنال‌های صوتی مجموعه تست، ویژگی استخراج می‌گردد. سپس با استفاده از مدل آوایی و مدل زبانی که در مرحله آموزش حاصل شدند و همچنین به کمک دیکشنری، فرآیند رمز گشایی انجام شده و سیگنال صوت به دنباله کلمات متناظر نگاشت می‌شود. لازم به توضیح است که رمز گشایی به فرآیندی گفته می‌شود که طی آن مشخص می‌گردد کدام دنباله کلمات به سیگنال ورودی شباهت بیشتری دارد.

۱-۳- گام‌های اجرای پایان‌نامه

در این پایان‌نامه به‌منظور بهبود دقت و کارآیی سیستم‌های بازشناسی گفتار فارسی با استفاده از شبکه عصبی عمیق حافظه کوتاه مدت ماندگار و همچنین مجموعه دادگان فارس‌دات[۱] [۱۵] مدل آوایی برای بازشناسی گفتار فارسی ساخته شده است. این مجموعه شامل ۶۰۸۰ سیگنال صوتی فارسی با فرکانس نمونه برداری ۲۰ کیلو هرتز می‌باشد و در مجموع شامل ۳۸۶ جمله است که توسط ۳۰۰ فارسی زبان که این افراد متعلق به ده لهجه متفاوت هستند، خوانده شده است. در ادامه گام‌های پیاده‌سازی این پایان‌نامه به اختصار شرح داده می‌شود.

۱. استخراج ویژگی: به‌منظور استخراج ویژگی از سیگنال‌های صوتی از دو روش ضرایب کپسترال در مقیاس مل[۲] (MFCC) [۱۶] و شبکه باور عمیق[۳] (DBN) [۱۷] استفاده شده است. در روش MFCC از هر فریم ۳۹ ویژگی استخراج گردیده است و ویژگی‌های MFCC به‌دست آمده، ورودی شبکه DBN می‌باشد. همچنین به‌منظور مقابله با نویز هر بردار ویژگی MFCC نرمال‌سازی[۴] شده است. یعنی هر بردار ویژگی به بردار ویژگی با میانگین صفر و انحراف معیار یک تبدیل گردیده است.

---

[۱] Farsdat

[۲] Mel Frequency Cepstral Coefficient (MFCC)

[۳] Deep Belief Network (DBN)

[۴] Normalization

۲. آموزش: جهت مدل‌سازی آوایی، از شبکه‌های عصبی حافظه کوتاه مدت ماندگار یک‌طرفه، حافظه کوتاه مدت ماندگار دوطرفه و همچنین حافظه کوتاه مدت ماندگار عمیق یک‌طرفه و دوطرفه با لایه خروجی طبقه‌بند زمانی پیوندگرا[۱] (CTC) جهت تشخیص دنباله واج متناظر با سیگنال ورودی استفاده گردیده است. جزئیات این روش‌ها در فصل چهارم و همچنین نتایج مربوط به پیاده‌سازی در فصل پنجم شرح داده شده است.

۳. ارزیابی: به‌منظور ارزیابی شبکه از معیار ارزیابی نرخ خطای واج[۲](PER) و همچنین نرخ خطای فریم[۳] (FER) استفاده گردیده است. در روش نرخ خطای واج، نسبت تعداد واج‌های درست تشخیص داده شده به کل واج‌ها و در روش نرخ خطای فریم، نسبت تعداد فریم‌های درست تشخیص داده شده به کل فریم‌ها محاسبه می‌گردد.

۱-۴- نوآوری پایان‌نامه

نوآوری مورد نظر این پایان‌نامه به‌شرح زیر می‌باشد:

۱. استفاده از شبکه حافظه کوتاه مدت ماندگار دوطرفه: پیش از این در [۱۸, ۱۹] از شبکه حافظه کوتاه مدت ماندگار جهت بازشناسی گفتار فارسی استفاده شده است اما این کار فقط با شبکه یک‌طرفه انجام گردیده است. ما در این پایان‌نامه برای اولین بار از حالت دوطرفه این شبکه جهت بازشناسی گفتار فارسی استفاده کرده‌ایم که استفاده از این روش موجب بهبود دقت تشخیص واج به‌میزان ۲٫۳٪ نسبت به شبکه حافظه کوتاه مدت ماندگار یک‌طرفه شده است.

۲. استفاده از شبکه حافظه کوتاه مدت ماندگار عمیق: استفاده از شبکه عصبی عمیق یک‌طرفه، موجب بهبود دقت تشخیص واج به‌میزان ۳٫۳٪ نسبت به حالت یک لایه آن شده است. همچنین استفاده از شبکه عصبی عمیق دوطرفه موجب بهبود دقت تشخیص واج به‌میزان ۳٫۶٪ نسبت به شبکه دوطرفه یک لایه گردیده است.

۳. استفاده از شبکه باور عمیق جهت استخراج ویژگی: این امر موجب بهبود دقت تشخیص واج شبکه حافظه کوتاه مدت ماندگار غیر عمیق یک‌طرفه به میزان ۱٪ و بهبود دقت تشخیص واج شبکه عمیق حافظه کوتاه مدت ماندگار دوطرفه به‌میزان ۰٫۴٪ در مقایسه با ویژگی‌های MFCC شده است. همچنین دقت تشخیص فریم شبکه‌های حافظه کوتاه مدت ماندگار غیر عمیق یک‌طرفه و دوطرفه و شبکه عمیق حافظه کوتاه مدت ماندگار یک‌طرفه را حدود ۱٪ در مقایسه با ویژگی‌های MFCC بهبود داده است.

---

[۱] Connectionist Temporal Classification (CTC)

[۲] Phoneme Error Rate (PER)

[۳] Frame Error Rate (FER)

۱-۵- خلاصه فصل‌ها

در ادامه‌ی پایان‌نامه، در فصل دوم مروری بر پژوهش‌های پیشین حوزه بازشناسی گفتار در زبان انگلیسی و فارسی خواهیم داشت. پس از آن در فصل سوم انواع شبکه‌های عصبی را مورد بررسی قرار خواهیم داد. در فصل چهارم گام‌های پیاده‌سازی پایان‌نامه و الگوریتم‌های مورد استفاده به تفصیل شرح داده خواهد شد. سپس در فصل پنجم به توضیح جزئیات و نتایج پیاده‌سازی‌های انجام شده می‌پردازیم و در انتها در فصل ششم پیشنهاد برای کارهای آینده را ارائه خواهیم کرد.

فصل دوم: مروری بر پژوهش‌های پیشین

۲-۱- مقدمه

در این فصل ابتدا به کارهای انجام شده در زمینه بازشناسی گفتار زبان انگلیسی خواهیم پرداخت. پس از آن سیر تکاملی بازشناسی گفتار در زبان فارسی را مرور خواهیم کرد و در انتها مروری بر روند تکاملی شبکه‌های عصبی خواهیم داشت.

۲-۲- بررسی روند تکاملی بازشناسی گفتار انگلیسی

نخستین تلاش‌ها در حوزه بازشناسی گفتار مربوط به سال ۱۹۳۶ یعنی دهه ۴۰ میلادی می‌باشد که ایالات متحده آمریکا پروژه‌ای را برای ترجمه خودکار از زبان روسی به زبان انگلیسی تعریف کرد [۲۰]. از آنجایی‌که روش پیاده‌سازی این پروژه یک روش بالا به پایین[۱] بود، این پروژه منجر به شکست شد. در روش بالا به پایین ابتدا کلمات سیگنال صوتی تشخیص داده می‌شوند سپس از اتصال این کلمات به یکدیگر متن معادل سیگنال صوتی استخراج می‌گردد. در حالی‌که روش مورد استفاده در سیستم‌های بازشناسی گفتار امروزی پایین به بالا[۲] می‌باشد، یعنی در گام نخست واحدهای آوایی تشخیص داده می‌شوند، سپس در گام بعدی کلمات و در آخرین مرحله جمله مشخص می‌گردد. تلاش بعدی برای طراحی

[۱] Top-Down

[۲] Down-Top

سیستم بازشناسی گفتار در دانشگاه کارنگی ملون[۱] (CMU) ایالات متحده انجام گردید که هدف این پروژه طراحی سیستمی برای تشخیص عبارت‌های امری در حرکات شطرنج بود [۲۰]. ولی با توجه به این‌که روش پیاده‌سازی آن مانند پروژه قبلی برگرفته از روش بالا به پایین بود، این پروژه نیز نتیجه‌ای در پی نداشت. پس از آن در سال ۱۹۵۲ یعنی اوایل دهه ۵۰ میلادی در آزمایشگاه بل[۲]، سیستم تشخیص اعداد گسسته برای یک گوینده طراحی گردید [۲۱] که این سیستم بر مبنای اندازه‌گیری فرکانس‌های تشدید گفتار (فرمنت‌ها) در حوزه حروف صدادار هر عدد کار می‌کرد. در سال ۱۹۵۶ در آزمایشگاه RCA تلاش برای ساخت سیستمی جهت بازشناسی ۱۰ هجای مختلف برای یک گوینده انجام شد [۲۲] که این ده هجا از ده کلمه تک هجایی انتخاب گردیده بودند. مبنای کار این سیستم اندازه‌گیری‌های طیفی در حوزه حروف صدادار هر کلمه با استفاده از یک بانک فیلتر آنالوگ بود. آخرین تلاش برای طراحی سیستم بازشناسی گفتار در دهه ۵۰ میلادی در سال ۱۹۵۹ توسط فری و دنز[۳] در کالج انگلستان انجام گردید که در این پروژه سیستمی جهت تشخیص واج‌ها در سطح ۹ واج بی‌صدا و ۴ واج صداردار با استفاده از یک تحلیل کننده طیفی[۴] و همچنین یک انطباق دهنده الگوی طیفی طراحی گردید [۲۳]. در این پروژه جهت بهبود دقت در بازشناسی واج‌ها برای کلمات دو یا چند واجی از اطلاعات آماری در مورد رشته‌های واجی مجاز در زبان انگلیسی استفاده گردید. در دهه ۶۰ میلادی تلاش‌های متعددی جهت ساخت سخت‌افزارهای خاص منظوره مرتبط با بازشناسی گفتار انجام گردید که از جمله آن می‌توان به طراحی سخت افزار شناسایی واج در دانشگاه کیوتو[۵] در سال ۱۹۶۲ [۲۴] و همچنین ساخت سخت‌افزار مربوط به شناسایی ارقام در آزمایشگاه NEC در سال ۱۹۶۳ اشاره کرد [۲۵]. در اواخر دهه ۶۰ مجموعه‌ای از روش‌های نرمال‌سازی زمانی[۶] جهت تعیین محل ابتدا و انتهای گفتار در سیگنال صوتی در دانشگاه RCA ارائه گردید [۲۶]. در سال ۱۹۶۸ روش انطباق زمانی پویا[۷] (DTW) در شوروی سابق ارائه گردید که این روش بر مبنای برنامه‌سازی پویا[۸] (DP) می‌باشد [۲۷]. از دیگر دستاوردهای مهم اواخر دهه ۶۰ تحقیقات ردی[۹] در زمینه بازشناسی گفتار پیوسته در تعقیب دینامیک واج‌ها بود که منجر به تحقیقات طولانی و دنباله‌داری در این حوزه در دانشگاه CMU گردید [۲۸] به‌طوریکه این دانشگاه تا به امروز در زمینه طراحی سیستم‌های بازشناسی گفتار پیوسته پیش‌رو می‌باشد.

---

[۱] Carnegie Mellon University

[۲] Bell Labs

[۳] Fry and Dense

[۴] Spectral Analyzer

[۵] Kyoto

[۶] Time Normalization

[۷] Dynamic Time Warping (DTW)

[۸] Dynamic Programming (DP)

[۹] Reddy

در دهه هفتاد موفقیت‌های بزرگی پدیدار گشت که در این میان روس‌ها و ژاپنی‌ها پیشگام بودند. ایتاکورا[۱] استفاده از رمز گذاری پیش‌بینی خطی[۲] (LPC) را که در ساخت رمزکننده‌ها با نرخ پایین موفق بود، در بازشناسی گفتار مطرح کرد و برای این کاربرد معیار فاصله مناسبی را بیان کرد [۲۹]. همچنین در این دهه شرکت IBM تحقیقات گسترده و دنباله‌داری را در زمینه بازشناسی گفتار بر روی مجموعه دادگان بزرگ آغاز نمود [۳۰]. در همین زمان تلاش‌هایی جهت مستقل از گوینده کردن سیستم‌های بازشناسی گفتار در آزمایشگاه‌های بل انجام گردید [۳۱]. یکی دیگر از فعالیت‌های شاخص این دهه ارائه سیستم درک گفتار Harpy در سال ۱۹۷۳ توسط CMU بود [۳۲] که تحت پروژه DARPA[۳] انجام گردید.

برخلاف دهه ۷۰ میلادی تحقیقات به‌طور عمده روی بازشناسی گفتار گسسته متمرکز بود، در دهه ۸۰ میلادی تمرکز عمده بر روی طراحی سیستم‌های بازشناسی گفتار پیوسته بود. در دهه ۸۰ محققان بر روی طراحی سیستم‌هایی که بتوانند رشته‌ای از کلماتی را که با مکث ادا می‌شوند تشخیص دهند، تحقیق کردند. در این دهه طیف گسترده‌ای از الگوریتم‌های مبتنی بر تطبیق الگو برای بازشناسی کلمات متصل ارائه شد که از آن‌ها می‌توان به الگوریتم ارائه شده توسط ساکو[۴] با رویکرد برنامه نویسی پویا دو سطحی که در NEC پیاده‌سازی شد [۳۳] و همچنین الگوریتم با رویکرد ساخت سطح[۵] که در آزمایشگاه‌های بل ارائه شد [۳۴]، نام برد. تحقیقات در این دهه با انتقال از روش‌های تطابق الگو به روش‌های مبتنی بر مدل‌های آماری به‌ویژه مدل مخفی مارکوف همراه بود [۲, ۳۵]. اگرچه روش مدل مخفی مارکوف در سال ۱۹۸۰ معرفی شده بود ولی تا اواسط این دهه که تئوری مباحث مربوط به مدل مخففی مارکوف ارائه شد، مورد استفاده قرار نگرفت. در سال ۱۹۸۸ اولین سیستم بازشناسی گفتار پیوسته مستقل از گوینده موفق در دانشگاه CMU تحت عنوان SPHINX روی مجموعه واژگان بزرگ ساخته شد [۳۶] این سیستم نسخه بهبود یافته سیستم Harpy می‌باشد. در اواخر این دهه روش‌های مبتنی بر شبکه عصبی برای طراحی سیستم‌های بازشناسی گفتار معرفی گردید ولی به علت مشکلات فراوانی که شبکه‌های عصبی داشتند مورد توجه قرار نگرفتند [۳۷].

اوایل دهه ۹۰ میلادی روش‌های مختلفی جهت ترکیب مدل مخفی مارکوف و شبکه عصبی به‌منظور طراحی سیستم‌های بازشناسی گفتار با دقت بالاتر انجام گردید [۳۸]. همچنین در این دهه به طراحی سیستم‌های بازشناسی گفتار مقاوم به نویز توجه بسیاری شد که نتیجه آن ارائه روش‌های مقابله با نویز از جمله بازگشت خطی با بیشینه شباهت[۶] (MLLR) [۳۹] و ترکیب موازی مدل[۷] (PMC) [۴۰] بود. در دهه ۹۰ به‌تدریج از سیستم‌های بازشناسی گفتار در کاربردهای واقعی‌تر از جمله دیکته متون و کاربردهای تلفنی استفاده گردید و همچنین بازشناسی گفتار بیشتر مورد توجه محققان غیر

[۱] Itakura

[۲] Linear predictive coding (LPC)

[۳] Defense Advanced Research Projects Agency (DARPA)

[۴] Sakoe

[۵] Level-Building

[۶] Maximum Likelihood Linear Regression (MLLR)

[۷] Parallel Model Combination (PMC)

انگلیسی زبان قرار گرفت که از جمله آن می‌توان به تحقیقات جدی در زمینه بازشناسی گفتار پیوسته زبان فارسی در اواخر دهه ۹۰ میلادی اشاره کرد. در سال ۱۹۹۷ با معرفی شبکه عصبی حافظه کوتاه مدت ماندگار توسط هاکریتر و اشمیدبر[1] مشکل فراموشی دنباله‌های طولانی در شبکه‌های عصبی بازگشتی برطرف گردید [۱۰] و موجب شد که استفاده از شبکه‌های عصبی بازگشتی بار دیگر در طراحی سیستم‌های بازشناسی گفتار مورد توجه محققین قرار بگیرد. در این شبکه نرون‌های لایه پنهان با بلوک‌های حافظه[2] جایگزین شده‌اند.

در سال ۲۰۰۱ شبکه حافظه کوتاه مدت ماندگار توسط فلیکس گرز[3] توسعه داده شد [۹] و به ساختار بلوک حافظه دروازه فراموشی نیز اضافه شد. در سال ۲۰۰۲ برنامه EARS[4] با تاکید بر دستیابی به دقت و کارایی بالاتر سیستم‌های بازشناسی توسط DARPA معرفی گردید [۴۱] که هدف اصلی آن افزایش توانایی این سیستم‌ها برای صحبت کردن به‌صورت طبیعی، تشخیص مرز جملات، تبدیل به متن کردن گفتارهای مربوط به چند زبان است. با توجه به این‌که سیستم‌های بازشناسی گفتار جهت آموزش نیاز به سیگنال‌های صوتی رونویسی شده دارند و این کاری بسیار زمان‌بر و سخت می‌باشد، به‌همین منظور در سال ۲۰۰۲ جوزپه ریکاردی[5] روشی جهت کاهش سختی رونویسی سیگنال‌های صوتی با استفاده از ارائه الگوریتم یادگیری فعال[6] ارائه کرد و تعداد نمونه‌های آموزشی که باید برچسب گذاری شوند را با انتخاب نمونه‌هایی که حاوی بیشترین اطلاعات هستند کاهش داد [۴۲]. در سال ۲۰۰۵ الکس گریوز[7] شبکه عصبی حافظه کوتاه مدت ماندگار با ساختار دوطرفه را معرفی کرد که در مقایسه با حالت یک‌طرفه آن دقت بالاتری داشت [۱۱]. در سال ۲۰۰۶ با معرفی روش طبقه‌بند زمانی پیوندگرا توسط گریوز [۴۳]، این امکان فراهم شد تا شبکه‌های عصبی به‌طور مستقل برای طراحی سیستم‌های بازشناسی گفتار مورد استفاده قرار بگیرند. در این روش شبکه یاد می‌گیرد که به‌طور مستقیم یک نگاشت از سیگنال صوت به دنباله واج متناظر انجام دهد. در مقاله‌ای که در سال ۲۰۱۱ توسط لی دنگ[8] و دانگ یو[9] منتشر گردید، جهت بازشناسی گفتار پیوسته با واژگان بزرگ یک مدل وابسته به بافت با ترکیب شبکه باور عمیق و مدل مخفی مارکوف ارائه گردید [۴۴] که نتایج به‌دست آمده بسیار نوید بخش بود. در مقاله [۴۵] که در سال ۲۰۱۲ منتشر گردید از شبکه عصبی پیچشی[10] (CNN) در مدل ترکیبی شبکه عصبی و مدل مخفی مارکوف جهت تشخیص گفتار استفاده شد و نتایج به دست آمده نشان داد که استفاده از این شبکه عصبی به‌جای شبکه عصبی بازگشتی متداول روی مجموعه دادگان

[1] Hochreiter & Schmidhuber
[2] Memory Block
[3] Felix Gers
[4] Efficient Affordable Reusable Speech-to-Text (EARS)
[5] Giuseppe Richardi
[6] Active Learning
[7] Alex Graves
[8] Li Deng
[9] Dong Yu
[10] Convolutional Neural Networks (CNN)

TIMIT میزان خطا را به‌میزان قابل توجهی کاهش می‌دهد. پس از آن در سال ۲۰۱۳ شبکه عصبی عمیق دوطرفه حافظه کوتاه مدت ماندگار با لایه خروجی CTC جهت تشخیص واج را ارائه شد [۵] که منجر به بهبود قابل توجه نتایج تشخیص واج روی مجموعه دادگان TIMIT گردید. پس از آن در مقاله‌ای که در سال ۲۰۱۵ منتشر گردید، از ترکیب شبکه عصبی پیچشی و شبکه حافظه کوتاه مدت ماندگار با ساختار عمیق مدلی جهت بازشناسی گفتار در قالب یک ساختار واحد ارائه شد که منجر به بهبود دقت تشخیص کلمه در مقایسه با شبکه حافظه کوتاه مدت ماندگار گردید [۴۶]. در مقاله‌ای که در سال ۲۰۱۷ ارائه شد از ترکیب شبکه عصبی عمیق پیچشی بدون اتصالات بازگشتی و طبقه‌بند زمانی پیوندگرا جهت بازشناسی گفتار سر به سر[۱] استفاده شده است استفاده از این روش روی مجموعه دادگان TIMIT نشان داد که روش ارائه شده کارایی بالایی دارد و از از لحاظ محاسباتی کارآمد است [۴۷]. جدول ۲-۱ خلاصه مطالب ذکر شده در روند تکامل تحقیقات در حوزه بازشناسی گفتار انگلیسی را نشان می‌دهد.

جدول ۰-۱) تاریخچه بازشناسی گفتار انگلیسی

سال	موضوع	منابع
۱۹۵۲	طراحی سیستم تشخیص اعداد گسسته در آزمایشگاه بل	[۲۱]
۱۹۵۶	ساخت سیستم بازشناسی ۱۰ هجای مختلف برای یک گوینده در آزمایشگاه RCA	[۲۲]
۱۹۵۹	طراحی سیستم تشخیص واج در سطح ۹ واج بی‌صدا و ۴ واج صداردار در انگلستان	[۲۳]
۱۹۶۲	طراحی سخت افزار تشخیص واج در دانشگاه کیوتو	[۲۴]
۱۹۶۳	ساخت سخت‌افزار مربوط به تشخیص اعداد گسسته در آزمایشگاه NEC در ژاپن	[۲۵]
۱۹۶۴	ارائه مجموعه‌ای از روش‌های نرمال‌سازی زمانی در RCA	[۲۶]
۱۹۶۶	آغاز تحقیقات ردی در زمینه بازشناسی گفتار پیوسته در دانشگاه CMU	[۲۸]
۱۹۶۸	ارائه روش انطباق زمانی پویا در شوروی	[۲۷]
۱۹۷۱	آغاز تحقیقات IBM در زمینه بازشناسی گفتار بر روی مجموعه دادگان بزرگ	[۳۰]
۱۹۷۳	معرفی سیستم Harpy در دانشگاه CMU تحت پروژه DARPA	[۳۲]

[۱] End-To-End

۱۹۷۵	معرفی LPC توسط ایتاکورا در بازشناسی گفتار	[۲۹]
۱۹۷۹	کار روی سیستم بازشناسی گفتار مستقل از گوینده در آزمایشگاه بل	[۳۱]
۱۹۷۹	برنامه نویسی پویای دو سطحی برای تشخیص کلمات متصل	[۳۳]
۱۹۸۰	معرفی مدل مخفی مارکوف	[۳۵ ,۲]
۱۹۸۱	معرفی روش level-building توسط آزمایشگاه بل	[۳۴]
۱۹۸۸	ارائه سیستم بازشناسی گفتار پیوسته تحت عنوان SPHINX در دانشگاه CMU	[۳۶]
۱۹۸۹	استفاده از شبکه عصبی جهت بازشناسی گفتار	[۳۷]
دهه ۹۰	ترکیب مدل مخفی مارکوف و شبکه عصبی برای بازشناسی گفتار	[۳۸]
۱۹۹۳	معرفی روش مقاوم‌سازی PMC	[۴۰]
۱۹۹۵	معرفی روش مقاوم‌سازی MLLR	[۳۹]
۱۹۹۷	معرفی شبکه عصبی بازگشتی حافظه کوتاه مدت ماندگار	[۱۰]
۲۰۰۱	توسعه شبکه حافظه کوتاه مدت ماندگار	[۹]
۲۰۰۲	معرفی برنامه EARS توسط DARPA	[۴۱]
۲۰۰۲	استفاده از یادگیری فعال در بازشناسی گفتار	[۴۲]
۲۰۰۵	معرفی شبکه عصبی BLSTM	[۱۱]
۲۰۰۶	ارائه الگوریتم CTC	[۴۳]
۲۰۱۱	بازشناسی گفتار پیوسته وابسته به بافت با استفاده از مدل ترکیبی DBN-HMM	[۴۴]
۲۰۱۲	بازشناسی گفتار با استفاده از شبکه عصبی CNN در مدل ترکیبی NN-HMM	[۴۵]
۲۰۱۳	استفاده از DBLSTM با لایه خروجی CTC جهت بازشناسی گفتار	[۶]
۲۰۱۵	استفاده از مدل ترکیبی شبکه‌های CNN و LSTM با ساختار عمیق جهت بازشناسی گفتار	[۴۶]

۲۰۱۷	استفاده از شبکه عصبی پیچشی عمیق و CTC جهت بازشناسی گفتار	[۴۷]

۲-۳- بررسی روند تکاملی بازشناسی گفتار فارسی

آغاز فعالیت‌ها برای طراحی سیستم بازشناسی گفتار فارسی مصادف با اویل دهه ۷۰ شمسی می‌باشد و این دوره مصادف با زمانی است که تحقیقات انجام گرفته در بازشناسی گفتار انگلیسی بیشتر به حالت کاربردی درآمده بود. بنابراین می‌توان گفت که بازشناسی گفتار در زبان فارسی قدمتی کمتر از ۳۰ سال دارد. در حال حاضر گروه‌های فعال در زمینه بازشناسی گفتار فارسی عمدتا دانشگاهیان در دانشگاه‌هایی مانند صنعتی شریف، تهران و امیرکبیر هستند. همچنین از دیگر گروه‌های فعال در این حوزه پژوهشکده پردازش هوشمند علائم[۱] (RCISP) و شرکت عصر گویش پرداز[۲] (AGP) می‌باشد. حامیان طرح‌های پژوهشی بازشناسی گفتار فارسی عمدتا مرکز تحقیقات مخابرات ایران[۳] (ITRC) و مرکز صنایع نوین وابسته به وزارت صنایع و معادن می‌باشد.

یکی از مهمترین چالش‌ها برای طراحی سیستم بازشناسی گفتار فارسی قبل از دهه ۷۰، عدم وجود دادگان گفتاری مناسب بود که موجب شده بود کارهای انجام شده بسیار ساده و با مجموعه دادگان محدود باشد. با تاسیس پژوهشکده پردازش هوشمند علائم و ارائه مجموعه دادگان گفتاری فارس‌دات برای محیط عادی [۱۵] در سال ۱۳۷۵ و مجموعه دادگان تلفنی[۴] [۴۸] در سال ۱۳۷۸ توسط این پژوهشکده زمینه برای تحقیقات جدی‌تر در این حوزه فراهم گردید. این پژوهشکده در سال ۱۳۷۸ نسخه اولیه سیستم بازشناسی گفتار شنوا [۴۹] را که مبتنی بر شبکه عصبی است ارائه کرد. این نسخه برای واژگان متوسط (حدود ۱۰۰۰ واژه) به‌همراه یک بانک کلمات با حجم حدود ۱۲۰۰۰کلمه پیاده‌سازی شده است. دقت این سیستم برای بازشناسی کلمات متصل برابر ۹۰٪ و همچنین برای بازشناسی گفتار پیوسته معادل ۶۰٪ بود. پس از آن در سال ۱۳۸۳ نسخه دوم سیستم شنوا توسط این پژوهشکده ارائه گردید [۵۰] که در این نسخه دقت سیستم برای بازشناسی کلمات متصل با ۳ درصد افزایش به ۹۳٪ و همچنین برای بازشناسی گفتار پیوسته با ۸ درصد بهبود به عدد ۶۸٪ رسید. از دیگر فعالیت‌های مهم این پژوهشگاه تهیه دادگان فارس‌دات بزرگ [۵۱] در سال ۱۳۸۳ و همچنین تهیه پیکره متنی زبان فارسی [۵۲] و فارس‌دات بزرگ تلفنی [۵۳] در سال ۱۳۸۵ اشاره کرد.

---

[۱] Research Center of Intelligent Signal Processing (RCISP)

[۲] Asr Gooyesh Pardaz (AGP)

[۳] Iran Telecommunication Research Center (ITRC)

[۴] TFarsDat

شرکت عصر گویش پرداز که در سال ۱۳۸۲ با همکاری جمعی از دانشجویان کارشناسی ارشد و دکترا دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف شکل گرفت، در سال ۱۳۸۳ نسخه اولیه سیستم بازشناسی گفتار نویسا [۵۴] را که مبتنی بر مدل مخفی مارکوف بود، عرضه کرد. در این نسخه بازشناسی گفتار پیوسته در محیط‌های عادی با مجموعه دادگان حدود هزار کلمه دقتی معادل ۷۵٪ داشت. پس از آن در سال ۱۳۸۵ نسخه دوم نویسا [۵۵] با دقت معادل ۹۵٪ در بازشناسی گفتار پیوسته مستقل از گوینده با مجموعه دادگان ۲۱۰۰۰ کلمه ارائه گردید. این سیستم تا به امروز بهترین سیستم بازشناسی گفتار زبان فارسی می‌باشد که به صورت تجاری عرضه می‌شود. همچنین نرم افزار مترجم گفتار به گفتار و همچنین نرم افزار تایپ گفتار زبان فارسی از دیگر محصولات این شرکت می‌باشد.

از دیگر فعالیت‌های حوزه بازشناسی گفتار در اواخر دهه ۸۰ ارائه نسخه سوم سیستم شنوا ۳ [۵۶] توسط RCISP در سال ۱۳۸۷می‌باشد. این سیستم در شرایطی که گوینده در محیطی آرام با لحن کتابی صحبت کند دقتی معادل ۹۰٪ دارد.

شرکت عصر گویش پرداز نسخه‌های تخصصی سیستم نویسا [۵۵] در حوزه‌های تخصصی مانند پزشکی، حقوقی و اسلامی را نیز عرضه کرده است و در سال ۱۳۹۴ نسل دوم تشخیص گفتار فارسی را به‌صورت وب سرویس ارائه کرده است. در ادامه تعدادی از پژوهش‌هایی که توسط دانشگاهیان در حوزه بازشناسی گفتار فارسی انجام شده است را معرفی می‌کنیم.

در [۵۷] از شبکه عصبی دوسویه به‌منظور غلبه بر نویز در سیستم‌های بازشناسی گفتار استفاده شده است. نتایج به‌دست آمده نشان می‌دهد که استفاده از شبکه‌های دوسویه راه حل مناسبی برای غلبه بر نویز است که این تأثیر ناشی از تشکیل و ظهور جاذب‌ها در شبکه دوسویه است که در مراحل تعلیم به شبکه آموزش داده می‌شوند و در تعامل شبکه جلوسو با شبکه معکوس نمود عینی پیدا می‌کنند.

در [۵۸] برای بازسازی بخش‌های از دست رفته طیف گفتار روشی ارائه شده است که در آن برای بازسازی یک فریم خاص از مؤلفه‌های طیفی سالم همان فریم و همچنین مؤلفه‌های سالم فریم‌های قبل و بعد از آن نیز استفاده می‌نماید. برای این منظور ابتدا با استفاده از HMM برای داده‌های تمیز مدل ساخته می‌شود. سپس با استفاده از این مدل و احتمال حضور مؤلفه‌های از دست رفته در حالت‌های مختلف، برای مؤلفه‌ی از دست رفته توزیع به‌دست می‌آید و در انتها با اعمال تخمین بیشترین احتمال پسین[۱] (MAP) بر آن، تخمینی برای مؤلفه از دست رفته حاصل می‌شود.

[۱] Maximum a Posteriori (MAP)

در [۵۹] یک مدل زبانی ترکیبی به‌منظور بهبود عملکرد سیستم‌های بازشناسی گفتار ارائه گردیده است. در این پژوهش مدل‌های موضوع پنهان[۱] (LTP) تحلیل معنایی پنهان احتمالاتی[۲] (PLSA) و تخصیص دیریکله پنهان[۳] (LDA) با مدل n-gram که در بازشناسی گفتار به‌صورت گسترده مورد استفاده قرار می‌گیرند ترکیب شده‌اند.

در [۶۰] به‌منظور مقاوم‌سازی[۴] سیستم‌های بازشناسی گفتار و حذف اثرات نویز ایده بهسازی مبتنی بر بازشناسی مطرح گردیده است که بر مبنای آن هرگونه بهسازی در بخش پیش پردازش سیستم‌های بازشناسی، باعث بهبود دقت بازشناسی می‌گردد. الگوریتم‌های بهسازی یک دسته از روش‌های مقاوم‌سازی در حوزه بازشناسی گفتار می‌باشد. بر مبنای ایده بهسازی مبتنی بر بازشناسی، دو الگوریتم تفاضل طیفی مبتنی بر بیشینه شباهت[۵] (MLBSS) و تفاضل طیفی مبتنی بر بیشینه کردن فاصله مدل‌ها[۶] (MDMBSS) برای به‌کارگیری تفاضل طیفی چند بانده در بخش پیش پردازش سیستم‌های بازشناسی گفتار ارائه شده است که این الگویتم‌ها در مواجهه با نویزهای جمع‌شونده[۷] موثر می‌باشند. همچنین برای حل مشکل نویزهای پیچشی[۸] راه حلی برای ترکیب آن‌ها با تکنیک نرمال‌سازی میانگین کپسترال[۹] (CMN) ارائه گردیده است.

در [۶۱] سیستمی برای بازشناسی گفتار فی‌البداهه–محاوره‌ای فارسی و تبدیل آن به گفتار رسمی ارائه شده است که شامل دو مرحله اصلی می‌باشد. در مرحله اول بازشناسی گفتار فی‌البداهه–محاوره‌ای انجام می‌شود و در مرحله دوم، این نوع گفتار به گفتار رسمی تبدیل می‌شود که این مرحله نیز شامل سه گام است. در گام اول ابتدا باید کلماتی که به فرم محاوره‌ای بیان شده‌اند و این‌که این کلمات چگونه باید اصلاح شوند تشخیص داده شود، در گام دوم بخش‌هایی که باید اصلاح شوند، با استفاده از پارامترهای استخراج شده از صدای گوینده اصلی سنتز شوند و در گام آخر به‌منظور در اختیار داشتن گفتاری طبیعی و با کیفیت باید نسبت به هموارسازی دنباله واجی در کلمات تغییر یافته اقدام شود.

در [۶۲] یک مدل آکوستیکی جدید به نام مدل پنهان برنولی ناهمگون با زمان (TI-HBM) بر مبنای مدل مخفی مارکوف پیشنهاد شده است که در آن فرآیند انتقال حالت مارکف با یک فرآیند برنولی تعمیم یافته جایگزین می‌شود. همچنین برای آموزش مدل TI-HBM، یک روش مبتنی بر الگوریتم حداکثرسازی امید ریاضی[۱۰] (EM) ارائه گردیده است. نتایج به‌دست آمده نشان‌دهنده بهبود دقت بازشناسی واج در مقایسه با مدل مخفی مارکوف می‌باشد.

---

۱ Latent Topic Model (LTP)

۲ Probabilistic Latent Semantic Analysis (PLSA)

۳ Latent Dirrichlet Allocation (LDA)

۴ Robustness

۵ Maximum Likelihood-Based Spectral Subtraction (MLBSS)

۶ Model Distance Maximization Based Spectral Subtraction (MDMBSS)

۷ Additive Noise

۸ Convolutional Noise

۹ Cepstral Mean Normalization (CMN)

۱۰ Expectation Maximization (EM)

در [۶۳] از روش تصویر حافظ خصوصیات محلی[۱] (LPP) متمایزساز جهت بهبود نرخ بازشناسی گفتار در شرایط نویزی استفاده گردیده است. روش تصویر حافظ خصوصیات محلی یکی از روش‌های تبدیل ویژگی مبتنی بر خمینه می‌باشد و دارای دو نسخه خطی و غیرخطی است. استفاده از این روش منجر به استخراج ویژگی‌هایی می‌شود که نسبت به نویز مقاوم‌تر هستند. با استفاده از روش پیشنهاد شده درصد بازشناسی نسبت به سیستم پایه (ضرایب مل کپستروم) بهبود یافته است.

در [۶۴] به‌منظور مقاوم‌سازی سیستم‌های بازشناسی گفتار پیوسته از روش آموزش تمایزگرایانه به عنوان جایگزینی برای معیار بیشینه درست‌نمایی[۲] (ML) به‌همراه انتقال بردار ویژگی و همچنین آموزش تطبیقی با گوینده[۳] (SAT) برای آموزش سیستم استفاده شده است. همچنین از بردار سری تیلور[۴] (VTS) جهت مقاوم‌سازی سیستم نسبت به نویز استفاده گردیده است. استفاده از روش معرفی شده جهت آموزش و همچنین بردار سری تیلور بر روی داده‌های نویزی شده TIMIT موجب بهبود دقت سیستم گردیده است.

در [۸] با استفاده از شبکه عمیق پرسپترون چند لایه[۵] یک سیستم بازشناسی گفتار طراحی گردیده است و این سیستم با استفاده از مجموعه دادگان فارس‌دات کوچک مورد ارزیابی قرار گرفته است.

در [۶۵] یک سامانه تشخیص اصطلاحات گفتاری که یکی از راه‌های بازیابی اطلاعات است طراحی شده است. این سامانه شامل دو مرحله پردازش گفتار به وسیله بازشناسی گفتار و همچنین جست‌وجو برای یک پرسش در میان خروجی بازشناسی می‌باشد که در مرحله بازشناسی از بازشناسی گفتار پیوسته استفاده شده است.

در [۶۶] یک ساختار وابسته به بافت برای بازشناسی گفتار پیوسته ارائه شده است. در این پژوهش با درنظر گرفتن واحد آوایی سه واجی، واج‌های پیشین و پسین هر واج در مدل‌سازی دخالت داده می‌شود. برای این منظور سه واجی‌های مشابه از طریق الگوریتم خوشه‌بندی[۶] K-Means در یک خوشه قرار می‌گیرند. همچنین برای حل مشکل ناهمسانی ابعاد جهت خوشه‌بندی، از سه الگوریتم انطباق زمانی پویا، تبدیل فوریه[۷] (FT) و تحلیل مولفه‌های اصلی[۸] (PCA) استفاده شده است که نتایج به‌دست آمده برتری روش تبدیل فوریه نسبت به دو روش دیگر را نشان می‌دهد. پس از خوشه‌بندی،

[۱] Locality Preserving Projection (LPP)

[۲] Maximum Likelihood (ML)

[۳] Speaker Adaptive Training (SAT)

[۴] Vector Taylor Veries (VTS)

[۵] Deep Multi-Layer Perceptreon

[۶] Clustering

[۷] Fourier Transform (FT)

[۸] Principal Component Analysis (PCA)

خوشه‌های با داده‌های آموزشی کم با یکدیگر ادغام شده‌اند و واج‌های با داده‌های آموزشی زیاد به عنوان سه واجی‌های ویژه، در یک خوشه مستقل قرار گرفته‌اند.

در [۶۷] یک الگوریتم جهت ترکیب ویژگی‌های دامنه و فاز تبدیل فوریه سیگنال به‌منظور تشخیص گفتار زبان فارسی ارائه شده‌است. در این الگوریتم ویژگی‌های MFCC و تابع تاخیر گروهی تغییر یافته مل[۱] (MMGDF) با یکدیگر ترکیب شده‌اند و ویژگی جدیدی به‌دست آمده است.

در [۱۹] از شبکه عصبی حافظه کوتاه مدت ماندگار جهت بازشناسی گفتار زبان فارسی استفاده شده است که نتایج به‌دست آمده نشان‌دهنده کارایی بالاتر این شبکه در مقایسه با مدل مخفی مارکوف در تشخیص واج می‌باشد.

جدول ۲-۲ خلاصه مطالب ذکر شده در روند تکامل تحقیقات در حوزه بازشناسی گفتار زبان فارسی را نشان می‌دهد.

جدول ۲-۲) تاریخچه بازشناسی گفتار فارسی

سال	توضیح	منابع
۱۳۷۵	ارائه مجموعه دادگان گفتاری فارس‌دات توسط RCISP	[۱۵]
۱۳۷۸	ارائه مجموعه دادگان گفتار تلفنی توسط RCISP	[۴۸]
۱۳۷۸	ارائه نسخه اولیه سیستم بازشناسی گفتار شنوا	[۴۹]
۱۳۸۳	ارائه نسخه دوم سیستم بازشناسی گفتار شنوا	[۵۰]
۱۳۸۳	تهیه دادگان فارس‌دات بزرگ	[۵۱]
۱۳۸۳	نسخه اولیه سیستم بازشناسی گفتار نویسا	[۵۴]
۱۳۸۵	نسخه دوم سیستم بازشناسی گفتار نویسا	[۵۵]
۱۳۸۵	تهیه پیکره متنی زبان فارسی	[۵۲]
۱۳۸۵	تهیه فارس‌دات بزرگ تلفنی	[۵۳]

[۱] Mel-Modified Group Delay Function (MMGDF)

۱۳۸۶	کاربرد شبکه‌های عصبی دوسویه در تشخیص گفتار	[۵۷]
۱۳۸۷	نسخه سوم سیستم شنوا	[۵۶]
۱۳۸۸	بازشناسی مقاوم به نویز گفتار بر پایه روش‌های ویژگی گمشده	[۵۸]
۱۳۸۹	ارائه یک مدل زبانی ترکیبی برای بهبود عملکرد سیستم‌های بازشناسی گفتار پیوسته	[۵۹]
۱۳۸۹	مقاوم‌سازی سیستم‌های بازشناسی گفتار بر مبنای روش‌های جبران داده و تئوری ویژگی‌های گم‌شده	[۶۰]
۱۳۸۹	بازشناسی گفتار فی‌البداهه–محاوره‌ای و تبدیل آن به گفتار رسمی	[۶۱]
۱۳۸۹	بهبود مدل آکوستیکی مبتنی بر مدل پنهان مارکف	[۶۲]
۱۳۹۱	بهبود نرخ بازشناسی گفتار در شرایط نویزی با استفاده از روش‌های غیرخطی تبدیل ویژگی	[۶۳]
۱۳۹۲	مقاوم‌سازی سیستم بازشناسی گفتار پیوسته	[۶۴]
۱۳۹۲	یادگیری ژرف برای بازشناسی گفتار	[۸]
۱۳۹۳	طراحی و بهبود یک سامانه‌ی تشخیص اصطلاحات گفتاری	[۶۵]
۱۳۹۳	ارائه یک ساختار جدید وابسته به بافت برای بازشناسی گفتار پیوسته	[۶۶]
۱۳۹۴	استفاده همزمان از MFCC و اطلاعات فاز جهت تشخیص گفتار زبان فارسی	[۶۷]
۱۳۹۴	ارائه وب سرویس نویسا (نسل دوم بازشناسی گفتار فارسی)	[۵۵]
۱۳۹۵	بازشناسی گفتار فارسی با استفاده از شبکه حافظه کوتاه مدت ماندگار	[۱۹]

۲-۴- مروری بر روند تکاملی شبکه‌های عصبی

دهه ۴۰ میلادی دهه پیدایش شبکه‌های عصبی می‌باشد. در سال ۱۹۴۳ اولین نرون مصنوعی توسط وارن مک کلاچ[۱] و والتر پیتز[۲] نرون مک کلاچ-پیتز را معرفی شد [۶۸] و در سال ۱۹۴۷ توسعه داده شد. یکی از ویژگی‌های نرون مک کلاچ-پیتز مبتنی بر این ایده است که اگر ورودی شبکه به یک نرون از مقدار آستانه آن نرون بیشتر باشد، آن نرون برانگیخته می‌شود که این ایده امروزه در بسیاری از شبکه‌های عصبی مورد استفاده قرار می‌گیرد. پس از آن دونالد هب[۳] که از روانشناسان دانشگاه مک گیل[۴] بود اولین قانون یادگیری برای شبکه‌های عصبی را در سال ۱۹۴۹ معرفی کرد [۶۹]. ایده اصلی قانون یادگیری هب[۵] مبتنی بر این ایده بود که اگر دو نرون به‌طور هم‌زمان برانگیخته شوند استحکام اتصال بین آن‌ها باید افزایش یابد.

دهه ۵۰ و ۶۰ اولین عصر طلایی شبکه‌های عصبی است. در سال ۱۹۵۸ شبکه عصبی پرسپترون[۶] توسط فرانک روزنبلات[۷] معرفی گردید [۷۰] و در سال‌های ۱۹۵۹ و ۱۹۶۲ بهبود داده شد [۷۱, ۷۲]. این شبکه متشکل از یک لایه ورودی (الهام گرفته شده از شبکه چشم) بود که با مسیرهایی وزن‌دار به نرون‌های پیوند دهنده متصل می‌شد و این وزن‌های قابل تنظیم‌کردن بود. قانون یادگیری پرسپترون[۸] که از روشی تکرار شونده برای تنظیم وزن‌ها استفاده می‌کند، از قانون هب بسیار قوی‌تر است. در سال ۱۹۶۰ برنارد ویدرو[۹] و دانشجوی آن تد هاف[۱۰] قانون یادگیری آدالاین[۱۱] را معرفی کردند [۷۳] که با قانون یادگیری پرسپترون ارتباط تنگاتنگی دارد و با نام قانون دلتا[۱۲] یا ویدرو-هاف[۱۳] و یا میانگین مربعات کمینه[۱۴] (LMS) شناخته می‌شود. در قانون پرسپترون هر نرونی که پاسخ نادرست داشته باشد وزن‌های متصل به آن بروز

---

۱ Warren McCulloch

۲ Walter Pitts

۳ Donald Hebb

۴ McGill University

۵ Hebbian Learning

۶ Perceptron Neural Network

۷ Frank Rosenblatt

۸ Perceptron Learning

۹ Bernard Widrow

۱۰ Ted Hoff

۱۱ Adaline Learning

۱۲ Delta Rule

۱۳ Widrow-Hoff Rule

۱۴ Least Mean Square (LMS)

رسانی می‌شوند ولی در قانون دلتا وزن‌ها به‌گونه‌ای تنظیم می‌شوند که اختلاف بین خروجی شبکه و خروجی هدف کاهش یابد. قانون دلتا منجر به افزایش قابلیت تعمیم[۱] شبکه می‌گردد، بدین معنی که شبکه می‌تواند به ورودی‌های که مشابه داده‌های آموزشی هستند به درستی پاسخ دهد. به این شبکه آدالاین گفته می‌شود و می‌توان آن‌را نرون خطی وفقی[۲] یا سیستم خطی وفقی[۳] تفسیر کرد. قانون دلتا برای شبکه یک لایه، مبنای قانون پس انتشار[۴] برای شبکه‌های چند لایه می‌باشد. در سال ۱۹۶۹ مینسکی و پاپرت[۵] محدودیت‌هایی را برای پرسپترون بیان کردند.

دهه ۱۹۷۰ سال‌های خاموش شبکه‌های عصبی بود که علت این نام‌گذاری عمدتا به دلیل عدم موفقیت پرسپترون یک لایه در حل مسائلی مانند تابع XOR و همچنین عدم وجود روش کلی برای شبکه‌های چند لایه است. در سال ۱۹۷۲ اولین کار کوهنن[۶] که عضو دانشگاه هلسینکی[۷] بود روی شبکه عصبی حافظه پیوندی انجام شد [۷۴]. همچنین در سال ۱۹۷۲ جیمز آندرسون[۸] عضو دانشگاه براون[۹]، تحقیق در زمینه شبکه‌های عصبی حافظه انجمنی[۱۰] را آغاز کرد و نظریاتش را در سال ۱۹۹۷ در اثر خود با نام "حالت مغز در یک جعبه[۱۱]" ارائه کرد [۷۵].

دهه ۸۰ میلادی دهه شکوفایی شبکه‌های عصبی است. در سال ۱۹۸۲ کوهنن نگاشت خودسازمانده[۱۲] (SOM) را توسعه داد [۷۶] که در آن از ساختارهای توپولوژیکی در خوشه‌بندی برای واحدهای خوشه استفاده کرد. از این شبکه برای بازشناسی گفتار کلمات فنلاندی و ژاپنی و همچنین حل مساله فروشنده دوره‌گرد[۱۳] استفاده شد. پس از آن در سال ۱۹۸۳ شبکه‌های عصبی ماشین بولتزمن[۱۴] توسط هینتون[۱۵] ارائه شد [۷۷]. در این شبکه‌ها جابه‌جایی بین واحدها بر اساس احتمال

---

۱ Generalization
۲ Adaptive Linear Neuron
۳ Adaptive Linear System
۴ Backpropagation
۵ Minsky & Papert
۶ Kohonen
۷ Helsinki
۸ James Anderson
۹ Brown University
۱۰ Associative Memory
۱۱ Brain-State-in-a-Box
۱۲ Self-Organizing Map (SOM)
۱۳ Traveling Salesman Problem
۱۴ Boltzman Machine
۱۵ Hinton

انتقال صورت می‌گیرد. همچنین در آن‌ها از ایده‌هایی همچون شبیه‌سازی سرد شدن تدریجی[۱] و همچنین تئوری تصمیم‌گیر بیز[۲] استفاده شده است. روش پس‌انتشار خطا توسط دیوید پارکر[۳] در سال ۱۹۸۵ [۷۸] و لی کان[۴] در سال ۱۹۸۶ [۷۹] به‌صورت جداگانه کشف شد. این الگوریتم پرکاربردترین الگوریتم آموزش پرسپترون‌های چند لایه است. از دیگر کارهای برجسته دهه ۸۰ ارائه شبکه‌های عصبی هاپفیلد[۵] [۸۰] توسط جان هاپفیلد به‌همراه دیوید تانک محقق AT&T بر اساس وزن‌های ثبات و همچنین فعال‌سازی وقفی می‌باشد که جزو شبکه‌های حافظه انجمنی محسوب می‌شوند و از آن‌ها در حل مسائل ارضای محدودیت همچون "مساله فروشنده دوره‌گرد" استفاده شده است. در سال ۱۹۸۷ گیل کارپنتر[۶] به‌همراه استفان گراس‌برگ[۷] نظریه نوسان وقفی[۸] (ART) را ارائه کرد [۸۱]. یک شکل از این شبکه‌ها با نام ART1 برای خوشه‌بندی بردارهای دودویی طراحی شده و شکل دیگر آن که ART2 نام دارد برای خوشه‌بندی بردارهای پیوسته ارائه شده است. یکی دیگر از فعالیت‌های دهه ۸۰ ارائه شبکه Neocognitron توسط فوکوشیما[۹] و همکارانش در آزمایشگاه‌های NHK در توکیو جهت بازشناسی نویسه‌ها می‌باشد [۸۲]. این شبکه مدل توسعه یافته شبکه خودسازمانده قدیمی‌تر با نام Cognitron است که در سال ۱۹۷۵ توسط فوکوشیما معرفی شد [۸۳] ولی قادر به بازشناسی نویسه‌هایی که مکان یا جهت آن‌ها جابه‌جا شده، نبود.

در سال ۱۹۹۰ شبکه عصبی المان[۱۰] توسط جفری المان معرفی شد [۸۴]. این شبکه که به آن شبکه عصبی بازگشتی ساده[۱۱] نیز گفته می‌شود و می‌توان از آن برای یادگیری دنباله‌ای از نویسه‌ها استفاده کرد. این شبکه را می‌توان یک شبکه "نسبتا بازگشتی" به‌شمار آورد که در آن اکثر اتصالات فقط پیش‌خور هستند. پس از آن در سال ۱۹۹۶ شبکه عصبی جردن[۱۲] توسط مایکل جردن معرفی شد [۸۵]. این شبکه بسیار به شبکه المان شبیه است با این تفاوت که دارای اتصالات

---

۱ Simulated Annealing

۲ Bayesian Decision Theory

۳ David Parker

۴ LeCun

۵ Hopfild Neural Network

۶ Gail Carpenter

۷ Stephen Grossberg

۸ Adaptive Resonance Theory (ART)

۹ Fukushima

۱۰ Elman Neural Network

۱۱ Simple Recurrent Network

۱۲ Jordan Neural Network

بازگشتی از لایه خروجی به لایه بافت[۱] می‌باشد. در سال ۱۹۹۷ شبکه عصبی بازگشتی دوطرفه[۲] توسط شوستر و پالیوال[۳] گردید [۸۶]. این شبکه، شامل دو لایه پنهان بازگشتی مجزا می‌باشد و مهمترین مزیت این شبکه‌ها نسبت به شبکه‌های یک‌طرفه این است که دنباله ورودی در دو جهت مختلف به شبکه داده می‌شود بنابراین خروجی شبکه در هر گام زمانی به کل دنباله ورودی وابسته خواهد بود. یکی دیگر از کارهایی که در سال ۱۹۹۷ انجام شد معرفی شبکه حافظه کوتاه مدت ماندگار توسط هاکریتر و اشمیدبر بود [۱۰]. LSTM یک شبکه عصبی بازگشتی می‌باشد که در آن نرون‌های لایه پنهان با بلوک‌های حافظه جایگزین شده‌اند که به‌دلیل وجود بلوک‌های حافظه، این شبکه می‌تواند دنباله‌های طولانی را یاد بگیرد. با معرفی این شبکه مشکل فراموشی شبکه‌های عصبی بازگشتی برطرف گردید.

در سال ۲۰۰۱ شبکه LSTM توسط فلیکس گرز توسعه داده شد و به ساختار بلوک حافظه دروازه فراموشی[۴] اضافه گردید [۹]. پس از آن در سال ۲۰۰۵ شبکه حافظه کوتاه مدت ماندگار دوطرفه توسط گریوز معرفی شد [۱۲]. این شبکه یک شبکه عصبی بازگشتی دوطرفه می‌باشد که در ساختار آن به‌جای نرون‌های دو لایه پنهان بلوک‌های حافظه شبکه LSTM قرار داده شده است و نتایج به‌دست آمده نشان می‌دهد این شبکه نسبت به شبکه یک‌طرفه کارایی بالاتری دارد. پس از آن در سال ۲۰۰۶ الگوریتم طبقه‌بند زمانی پیوندگرا توسط گریوز معرفی گردید [۴۳]. این الگوریتم این امکان را به شبکه LSTM می‌دهد که به‌جای برچسب گذاری هر فریم از سیگنال صوت دنباله واج متناظر با دنباله ورودی را تولید کند. بنابراین نیاز پس پردازش جهت تبدیل خروجی شبکه به دنباله واج را برطرف می‌کند.

دهه دوم قرن ۲۱ را می‌توان غلبه شبکه‌های عصبی بر روش‌های دیگر در حوزه هوش مصنوعی دانست که به لطف حجم زیاد داده و بهبود توان پردازشی رایانه‌ها، ابزارها هوشمندی بیشتری یافتند. در مقاله‌ای که در سال ۲۰۱۱ توسط لی دنگ و دانگ یو منتشر گردید، جهت بازشناسی گفتار پیوسته با واژگان بزرگ یک مدل وابسته به بافت با ترکیب شبکه باور عمیق و مدل مخفی مارکوف ارائه گردید [۴۴] که نتایج به‌دست آمده بسیار نوید بخش بود. با توجه به این‌که تعیین پارامترهای شبکه‌های عصبی عمیق از جمله نرخ یادگیری، تعداد لایه‌ها و همچنین تعداد نرون‌های لایه پنهان تاثیر به‌سزایی در کارایی آن‌ها دارد، در مقاله [۸۷] که در سال ۲۰۱۲ منتشر گردید با استفاده از روش بهینه‌سازی بیزی[۵] الگوریتمی جهت کاهش هزینه تعیین این پارامترها و تعیین دقیق‌تر آن‌ها ارائه شد. پس از آن در سال ۲۰۱۳ گریوز شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه را معرفی کرد [۶]. این شبکه که از روی هم قرار دادن لایه‌های پنهان شبکه حافظه کوتاه مدت ماندگار دوطرفه حاصل می‌گردد کارایی بسیار بالاتری در مقایسه با شبکه حافظه کوتاه مدت ماندگار

---

۱ Context Layer

۲ Bidirectional Recurrent Neural Network

۳ Schuster & Paliwal

۴ Forget Gate

۵ Bayesian Optimization

یک لایه دارد. در سال ۲۰۱۴ در مقاله‌ای که از دانشگاه تورنتو[۱] منتشر گردید، جهت حل مشکل بیش برازش[۲] در شبکه‌های عصبی عمیق الگوریتمی تحت عنوان "حذف تصادفی[۳]" معرفی گردید [۸۸] که ایده کلی آن بر مبنای حذف تصادفی نرون‌های لایه پنهان می‌باشد. یکی دیگر از کارهایی که در سال ۲۰۱۴ توسط دنگ انجام گرفت، استخراج ویژگی با استفاده از شبکه عصبی عمیق بود که در این روش مجموعه‌ای از بردارهای ویژگی از لایه خروجی و لایه‌های پنهان مختلف استخراج گردید و نتایج حاصل نشان داد که، بهترین بردار ویژگی مربوط به آخرین لایه پنهان می‌باشد و حتی نسبت به ویژگی‌های استخراج شده از لایه خروجی کیفیت بالاتری دارد [۸۹]. در مقاله‌ای که در سال ۲۰۱۵ منتشر گردید، از ترکیب شبکه عصبی پیچشی و شبکه حافظه کوتاه مدت ماندگار با ساختار عمیق مدلی جهت بازشناسی گفتار در قالب یک ساختار واحد ارائه گردید که منجر به بهبود دقت تشخیص کلمه در مقایسه با شبکه حافظه کوتاه مدت ماندگار گردید [۴۶]. همچنین در مقاله‌ای که در سال ۲۰۱۷ ارائه شد از ترکیب شبکه عصبی عمیق پیچشی بدون اتصالات بازگشتی و طبقه‌بند زمانی پیوندگرا جهت بازشناسی گفتار سر به سر[۴] استفاده شده است استفاده از این روش روی مجموعه دادگان TIMIT نشان داد که روش ارائه شده کارایی بالایی دارد و از از لحاظ محاسباتی کارآمد است [۴۷]. جدول۲-۳ خلاصه روند تکامل شبکه‌های عصبی مصنوعی را نمایش می‌دهد.

جدول ۰-۳) روند تکامل شبکه‌های عصبی

سال	توضیح	منابع
۱۹۴۳	معرفی اولین نرون مصنوعی توسط مک کلاچ و پیتز	[۶۸]
۱۹۴۹	معرفی قانون یادگیری هب توسط هب	[۶۹]
۱۹۵۸	معرفی شبکه عصبی پرسپترون توسط روزنبلات	[۷۰]
۱۹۶۰	معرفی قانون یادگیری آدالاین توسط ویدرو و هاف	[۷۳]
۱۹۷۲	معرفی شبکه عصبی حافظه انجمنی توسط کوهنن	[۷۴]
۱۹۷۷	اثر آندرسون با نام حالت مغز در یک جعبه	[۷۵]
۱۹۸۲	توسعه نگاشت خود سازمانده کوهنن	[۷۶]
۱۹۸۳	ارائه شبکه‌های عصبی ماشین بولتزمن	[۷۷]
۱۹۸۵	معرفی روش پس‌انتشار خطا توسط پارکر	[۷۸]
۱۹۸۶	معرفی مجدد روش پس‌انتشار خطا توسط لی کان	[۷۹]

[۱] University of Toronto

[۲] Over-Fitting

[۳] Drop-Out

[۴] End-To-End

۱۹۸۷	معرفی نظریه نوسان وفقی توسط کارپنتر و گراس	[۸۱]
دهه ۸۰	ارائه شبکه‌های عصبی هاپفیلد توسط هاپفیلد و تانک	[۸۰]
دهه ۸۰	ارائه شبکه Neocognitron توسط فوکوشیما	[۸۲]
۱۹۹۰	معرفی شبکه المان توسط المان	[۸۴]
۱۹۹۶	معرفی شبکه جردن توسط جردن	[۸۵]
۱۹۹۷	معرفی شبکه عصبی بازگشتی دوطرفه	[۸۶]
۱۹۹۷	معرفی شبکه LSTM توسط هاکریتر و اشمیدبر	[۱۰]
۲۰۰۱	توسعه شبکه LSTM توسط گرز	[۹]
۲۰۰۵	معرفی شبکه BLSTM توسط گریوز	[۱۲]
۲۰۰۶	معرفی الگوریتم CTC	[۴۳]
۲۰۱۱	بازشناسی گفتار پیوسته وابسته به بافت با استفاده از مدل ترکیبی DBN-HMM	[۴۴]
۲۰۱۲	استفاده از روش بهینه‌سازی بیزی جهت تعیین پارامترهای شبکه‌های عصبی عمیق	[۸۷]
۲۰۱۳	ارائه شبکه DBLSTM	[۶]
۲۰۱۴	ارائه روش حذف تصادفی جهت حل مشکل بیش برازش در شبکه‌های عمیق	[۸۸]
۲۰۱۴	استخراج ویژگی با استفاده از شبکه عصبی عمیق	[۸۹]
۲۰۱۵	استفاده از مدل ترکیبی شبکه‌های CNN و LSTM با ساختار عمیق جهت بازشناسی گفتار	[۴۶]
۲۰۱۷	استفاده از شبکه عصبی پیچشی عمیق و CTC جهت بازشناسی گفتار	[۴۷]

فصل سوم: مروری بر شبکه‌های عصبی

۳-۱- مقدمه

شبکه‌ی عصبی یک سیستم پردازش اطلاعات است که ویژگی‌های مشترکی با سیستم عصبی موجودات زنده دارد. هر شبکه عصبی حاوی تعدادی گره (نرون) می‌باشد که این گره‌ها از طریق یال‌های وزن‌دار به یکدیگر متصل شده‌اند. در واقع این وزن‌ها اطلاعات به‌کار گرفته شده در شبکه جهت حل مساله را نشان می‌دهند. هدف از آموزش شبکه، حل مساله‌ی مورد نظر می‌باشد. در این‌جا منظور از آموزش، به‌دست آوردن بهترین وزن‌ها در شبکه است. در این فصل به بحث در زمینه انواع شبکه‌های عصبی از جمله شبکه‌های عصبی پیش‌رو[۱] و بازگشتی می‌پردازیم.

۳-۲- شبکه‌های عصبی پیش‌رو

به شبکه‌های عصبی [۹۰-۹۲] که در ساختار آن‌ها اتصالات بین نرون‌ها تشکیل حلقه نمی‌دهد شبکه‌های عصبی پیش‌رو می‌گویند. در این شبکه‌ها اطلاعات تنها در جهت رو به جلو یعنی از لایه ورودی به سمت لایه خروجی حرکت

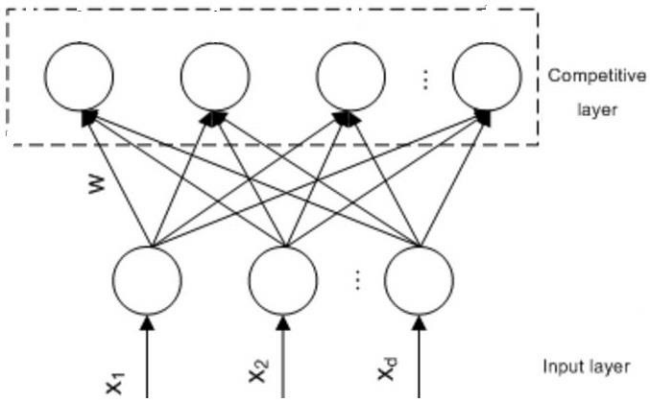
[۱] Feed Forward Neural Network

می‌کند. انواع مختلفی از این دسته از شبکه‌های عصبی وجود دارد که از جمله مشهورترین آن‌ها می‌توان به شبکه پرسپترون چندلایه[۱] (MLP) و شبکه نگاشت خود سازمانده کوهونن اشاره کرد. در ادامه به توضیح این شبکه‌ها می‌پردازیم.

۳-۲-۱- انواع شبکه‌های عصبی پیش‌رو

۳-۲-۱-۱- نگاشت خود سازمانده کوهونن

شبکه عصبی SOM [۹۰] برای حل مسائل خوشه‌بندی مورد استفاده قرار می‌گیرد. در این شبکه، تعداد نرون‌های خروجی را برابر تعداد خوشه‌ها و تعداد نرون‌های ورودی برابر ابعاد سیگنال ورودی در نظر می‌گیرند. همچنین فرض می‌شود که خوشه‌ها دارای آرایش یک بعدی یا دو بعدی منظم هستند. در طول فرآیند خود سازمانده، خوشه‌ای که بردار وزن آن کمترین فاصله را با بردار ورودی دارد به عنوان خوشه برنده در نظر گرفته می‌شود که در این فرآیند معمولا از فاصله اقلیدسی به عنوان معیار فاصله استفاده می‌گردد. در فرآیند آموزش شبکه، وزن‌های واحد برنده به‌همراه تمامی واحدهایی که در همسایگی مشخصی از آن قرار دارند به‌روز رسانی می‌گردد. شکل ۳-۱ ساختار شبکه SOM با آرایش خطی برای خوشه‌ها را نشان می‌دهد.



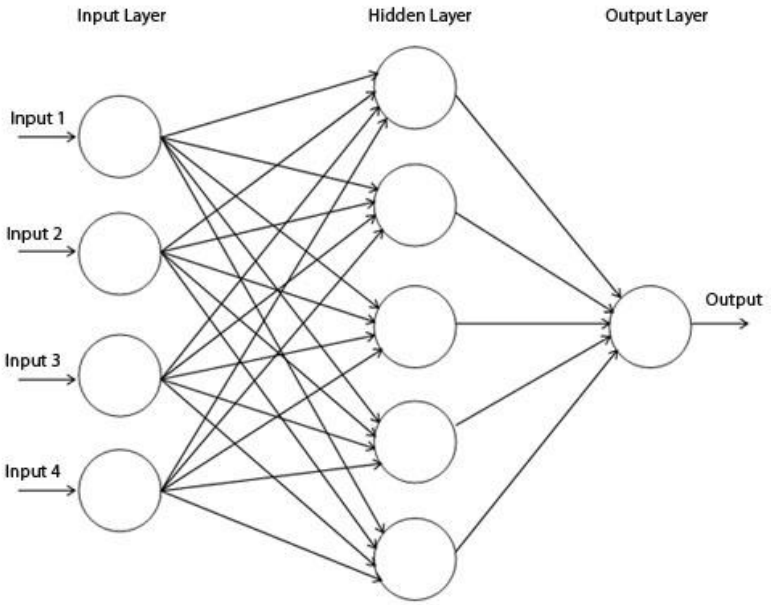
شکل ۳-۱) شبکه SOM با ساختار خوشه‌بندی خطی

۳-۲-۱-۲- شبکه عصبی پرسپترون چند لایه

شبکه پرسپترون چند لایه [۹۰] یک شبکه عصبی پیش‌رو می‌باشد که شامل سه لایه ورودی، پنهان و خروجی است. هدف از آموزش این شبکه رسیدن به قابلیت تعمیم و یادگیری می‌باشد. بدین معنی که شبکه بتواند به‌میزان قابل قبولی الگوهایی را که در مرحله آموزش ندیده است به‌درستی تشخیص دهد و همچنین الگوهای آموزش را نیز به‌درستی تشخیص

[۱] Multi-Layer Perceptron (MLP)

دهد. آموزش این شبکه در دو مرحله پیش‌رو و رو به عقب انجام می‌گیرد. بدین معنی که داده از لایه ورودی به سمت لایه خروجی حرکت می‌کند و پس از محاسبه خطا در لایه خروجی با استفاده از الگوریتم پس انتشار استاندارد این خطا از لایه خروجی به سمت لایه ورودی پس انتشار[1] می‌یابد. ساختار این شبکه در شکل ۳-۲ دیده می‌شود.



شکل ۳-۲) ساختار شبکه‌ی عصبی MLP

۳-۲-۱-۳- شبکه عصبی تاخیر زمانی

شبکه عصبی تاخیر زمانی[2] (TDNN) [۹۳] یک شبکه عصبی پیش‌رو می‌باشد که هدف اصلی آن کار بر روی داده‌های متوالی می‌باشد. به‌همین منظور از این شبکه در بازشناسی گفتار استفاده می‌گردد. اگر n_t تاخیر زمانی و به تعداد I نرون در لایه ورودی داشته باشیم، به هریک از نرون‌های ورودی $n_t + 1$ یال وارد می‌شود که تعداد n_t یال برای n_t ورودی قبلی و یک یال برای ورودی مربوط به گام زمانی فعلی می‌باشد. شکل ۳-۳ ساختار یک نرون تاخیر زمانی با n_t تاخیر برای هر نرون ورودی را نمایش می‌دهد. لازم به ذکر است که هر اتصال بین $z_i(t-t')$ و $z_i(t-t')+1$ وزنی معادل یک دارد. در ابتدا $z_i(t)$ به ازای $t=0$ مقداری غیر صفر دارد و $z_i(t-t')$ به‌ازای تمامی مقادیر i یعنی $i =$

[1] Back Propagate

[2] Time-Delay Neural Network (TDNN)

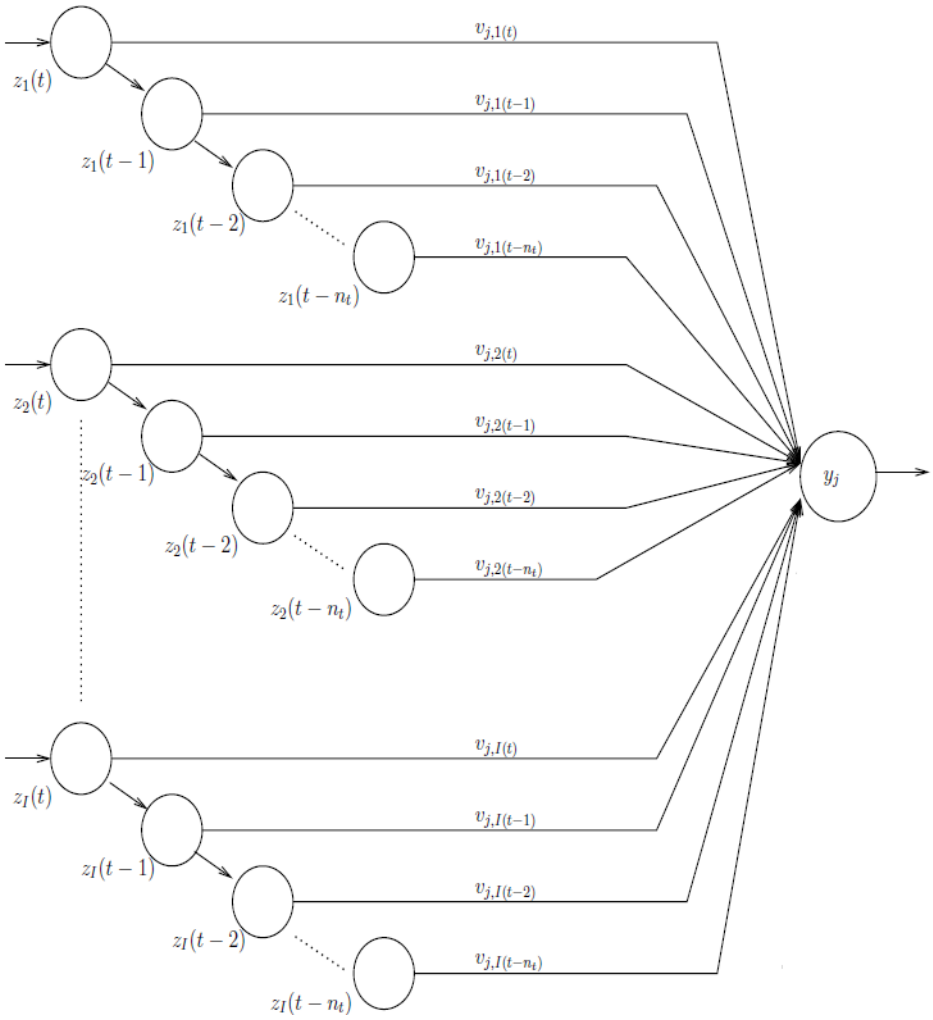
I .... .1 و تمامی گام‌های زمانی $t' = 1. \ldots . n_t$ مقدار صفر دارد. بلافاصله پس از این‌که ورودی اول به شبکه داده شد و قبل از ارائه ورودی دوم به شبکه داریم:

$$z_i(t-1) = z_i(t) \qquad (۱)$$

و به‌همین ترتیب پس از ارائه t' الگو به شبکه و قبل از ارائه الگوی $t' + 1$ داریم:

$$z_i(t-t') = z_i(t-t'+1) \qquad (۲)$$

که این امر باعث می‌شود در هر گام زمانی به‌روز رسانی وزن‌ها وابسته به الگوی فعلی و n_t الگوی قبلی باشد.



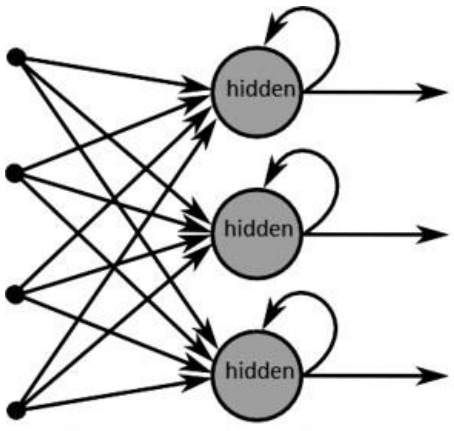
شکل ۳-۳) ساختار نرون تاخیر زمانی [۹۳]

۳-۳- شبکه‌های عصبی بازگشتی

شبکه‌های عصبی بازگشتی، به شبکه‌هایی گفته می‌شود که در ساختار آن‌ها یال‌های بازگشت کننده وجود دارد [۹, ۹۴]. شکل ۳-۴ ساختار یک شبکه‌های عصبی بازگشتی را نشان می‌دهد که در آن بازگشت محدود به لایه‌ی پنهان شده است. همان‌طور که دیده می‌شود هر نرون لایه پنهان از تمامی نرون‌های لایه ورودی و همچنین خروجی سایر نرون‌های لایه پنهان ورودی می‌گیرد.

مهمترین مزیت این شبکه‌ها نسبت به شبکه‌های پیش‌رو این است که، یک شبکه پیش‌رو مانند پرسپترون چند لایه تنها می‌تواند یک نگاشت بین بردارهای ورودی و بردارهای خروجی متناظرشان ایجاد کند، در حالی‌که شبکه‌های بازگشتی می‌توانند بین تمامی تاریخچه بردارهای ورودی قبلی به هر خروجی نگاشت ایجاد کنند. به‌عبارت دیگر مقدار بردار خروجی نه تنها به مقدار بردار ورودی فعلی وابسته است بلکه به بردارهای ورودی قبلی نیز وابستگی دارد و این ویژگی شبکه را قادر می‌سازد تا بتوند دنباله‌ای از ورودی‌ها را یاد بگیرد.

در ادامه به بررسی چند نوع از شبکه‌های عصبی بازگشتی از جمله شبکه هاپفیلد، المان و حافظه کوتاه مدت ماندگار می‌پردازیم.

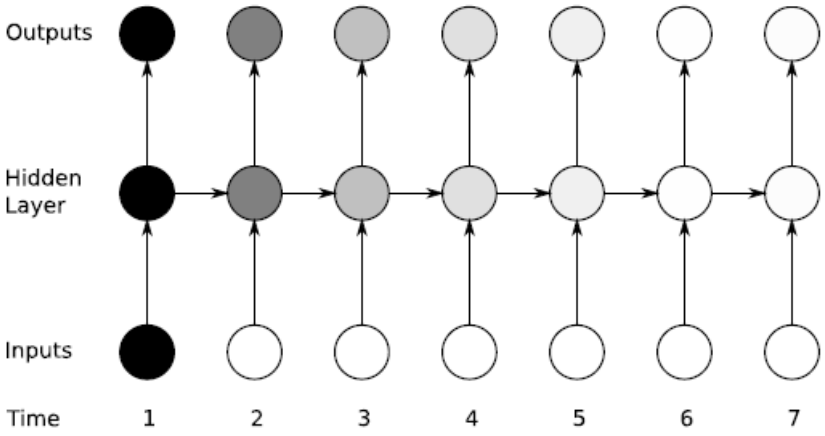


شکل ۳-۴) ساختار شبکه عصبی بازگشتی

۳-۳-۱- مشکل فراموشی دنباله‌های طولانی در شبکه‌های عصبی بازگشتی

داده‌های متوالی به داده‌هایی گفته می‌شود که در آن‌ها مقدار داده فعلی به مقدار داده‌های قبلی وابستگی دارد. یکی از بهترین نمونه‌های داده‌های متوالی سیگنال صوت است. ولی مشکل اصلی شبکه‌های عصبی بازگشتی برای یادگیری داده‌های متوالی این است که با طولانی شدن دنباله ورودی شبکه به مرور داده‌های اولیه را فراموش می‌کند [۹۴] و تاثیر

آن‌ها در خروجی شبکه به‌مرور کم‌رنگ‌تر می‌گردد. در نتیجه تعداد ورودی‌های مربوط به گام‌های زمانی قبلی که در عمل می‌توانیم به آن‌ها دسترسی داشته باشیم محدود است. بنابراین با توجه به حافظه محدود شبکه‌های عصبی بازگشتی نمی‌توان از آن‌ها برای یادگیری دنباله‌های طولانی استفاده کرد. همان‌طور که در شکل ۳–۵ دیده می‌شود، با طولانی شدن دنباله ورودی، شبکه کم‌کم ورودی‌های اولیه را فراموش می‌کند و تاثیر آن‌ها به‌مرور زمان با وارد شدن داده‌های جدید کم می‌شود. اولین داده ورودی، در گام زمانی یک بیشترین تاثیر را روی لایه‌های پنهان و خروجی دارد (سایه‌ها پررنگ تر هستند) ولی با گذشت زمان و وارد شدن داده‌های جدید تاثیر داده ورودی اول به مرور کمتر می‌شود. (سایه‌ها کمرنگ تر می‌شوند).



شکل ۳-۵) مشکل فراموشی شبکه‌های بازگشتی [۹۴]

۳–۳–۲– انواع شبکه‌های عصبی بازگشتی

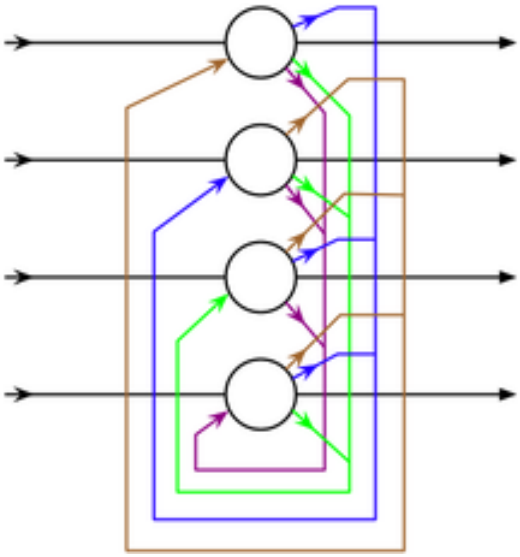
۳–۳–۲–۱– شبکه عصبی بازگشتی هاپفیلد

شبکه هاپفیلد که یک شبکه عصبی بازگشتی است، در سال‌های ۱۹۸۲ و ۱۹۸۴ توسط هاپفیلد ارائه شد [۹۰]. این شبکه یک شبکه کاملا به‌هم متصل می‌باشد و در آن هر واحد به تمامی واحدهای دیگر (به‌جز خودش) متصل است. همچنین وزن‌های این شبکه به صورت متقارن می‌باشد. در این شبکه هر نرون علاوه بر دریافت سیگنال از سایر نرون‌های شبکه، یک سیگنال خارجی که در واقع ورودی شبکه است را نیز دریافت می‌کند. (البته در نسخه اولیه این شبکه که در سال ۱۹۸۲ ارائه شد، شبکه سیگنال ورودی خارجی را فقط در اولین گام زمانی دریافت می‌کرد). این شبکه در هر مرحله تنها فعال‌ساز مربوط به یکی از نرون‌های خود را به کمک سیگنال دریافتی از سایر نرون‌ها و همچنین سیگنال خارجی وارد شده به آن واحد به‌روز می‌کند.

کاربرد اصلی شبکه هاپفیلد برای ذخیره‌سازی الگو و همچنین شناسایی الگوهای ذخیره شده در شبکه می‌باشد. در صورتی که الگو ورودی با یکی از الگوهای ذخیره شده در شبکه یکسان باشد، فعال‌سازهای واحدهای شبکه با الگوی ورودی یکسان می‌شوند و اگر بردار ورودی با هیچ‌یک از بردارهای ذخیره شده در شبکه یکسان نباشد، فعال‌سازهای واحدهای شبکه به مقادیری همگرا می‌شوند که با هیچکدام از بردارهای ذخیره شده در شبکه یکسان نیست. شکل ۳-۶ ساختار یک شبکه هاپفیلد با چهار نرون را نمایش می‌دهد.

برای ذخیره‌سازی P الگوی دودویی $s(p)$ در شبکه هاپفیلد که هر الگو به فرم $s(p) = \left(s_1(p). \ldots . s_n(p)\right)$ می‌باشد ماتریس وزن با استفاده از رابطه (۳) به‌دست می‌آید.

$$w_{ij} = \sum_{p} (2s_i(p) - 1)\left(2s_j(p) - 1\right). i \neq j \qquad (۳)$$



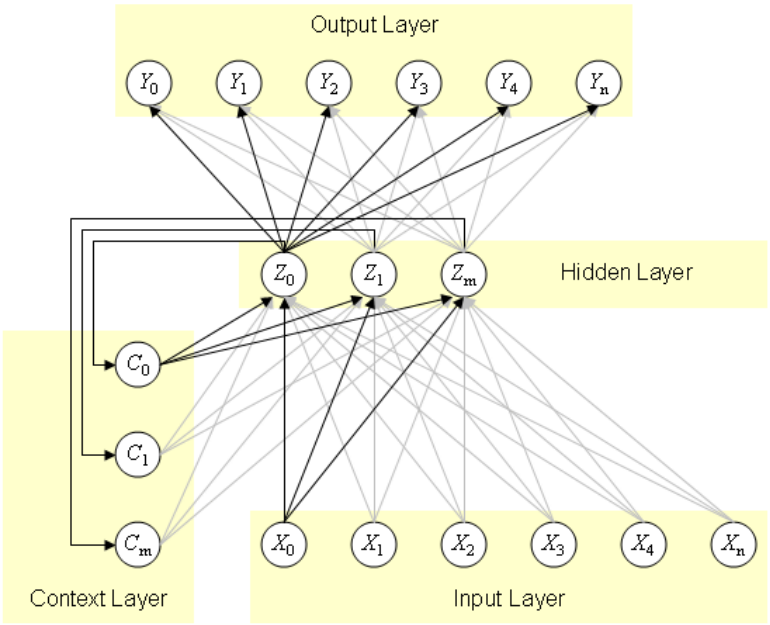
شکل ۳-۶) ساختار شبکه هاپفیلد با چهار نرون

۳-۳-۲-۲- شبکه عصبی المان

شبکه عصبی المان یک شبکه عصبی بازگشتی است که در سال ۱۹۹۰ توسط المان معرفی شد [۹۵]. این شبکه شامل چهار لایه ورودی، مخفی، بافت[۱] و همچنین یک لایه خروجی است. لایه ورودی به لایه پنهان متصل شده است و لایه پنهان علاوه بر این‌که به لایه خروجی متصل شده به لایه بافت نیز متصل می‌باشد. تعداد نرون‌های لایه بافت با

[۱] Context Layer

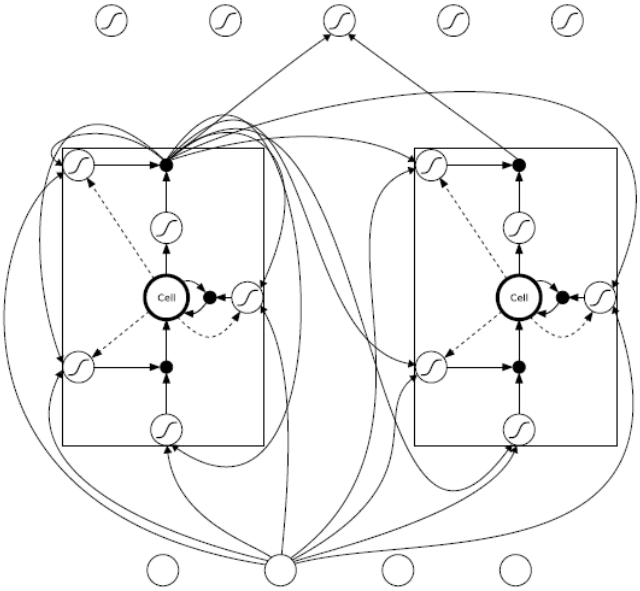
تعداد نرون‌های لایه پنهان برابر است، به‌علاوه وزن یال‌هایی که لایه پنهان را به لایه بافت متصل می‌کنند برابر مقدار ثابت یک می‌باشد. لازم به ذکر است که از هر نرون لایه بافت به تمامی نرون‌های لایه پنهان یال وجود دارد. در واقع اتصالات بازگشتی لایه بافت به لایه پنهان، یک حافظه کوتاه مدت را برای شبکه ایجاد می‌کند. به‌عبارت دیگر، نرون‌های لایه پنهان علاوه بر دریافت اطلاعات از ورودی فعلی شبکه، به کمک اتصالاتی که از لایه بافت به لایه پنهان وجود دارد اطلاعات مربوط به حالت قبلی نرون‌های لایه پنهان را نیز دریافت می‌کنند. شکل ۳-۷ ساختار شبکه المان را نمایش می‌دهد.



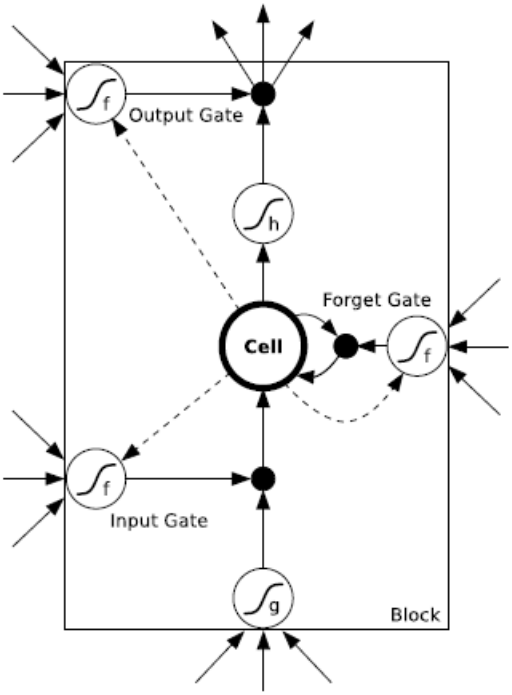
شکل ۳-۷) ساختار شبکه المان

۳-۳-۲-۳- شبکه عصبی حافظه کوتاه مدت ماندگار

در سال ۱۹۹۷ شبکه‌ی عصبی بازگشتی حافظه کوتاه مدت ماندگار برای اولین بار توسط هاکریتر و اشمیدبر معرفی شد [۱۰]. در این شبکه نرون‌های لایه پنهان با بلوک‌های حافظه جایگزین شدند که این امر باعث حل مشکل فراموشی دنباله‌های طولانی در شبکه‌های بازگشتی شد. در ساختار اولیه که در سال ۱۹۹۷ ارائه شد، هر بلوک حافظه شامل دو دروازه ورودی و خروجی بود ولی در سال ۲۰۰۱ شبکه حافظه کوتاه مدت ماندگار توسط فلیکس گرز توسعه داده شد [۹] و به ساختار بلوک حافظه دروازه فراموشی اضافه گردید. این دروازه، بلوک حافظه را قادر می‌سازد تا حالت فعلی خود را ریست نماید. شکل‌های ۳-۸ و ۳-۹ به‌ترتیب ساختار کلی این شبکه و بلوک حافظه را نمایش می‌دهد. در فصل چهارم این شبکه و انواع آن شرح داده خواهد شد.



شکل ۳-۸) ساختار شبکه LSTM با دو بلوک حافظه [۹۴]

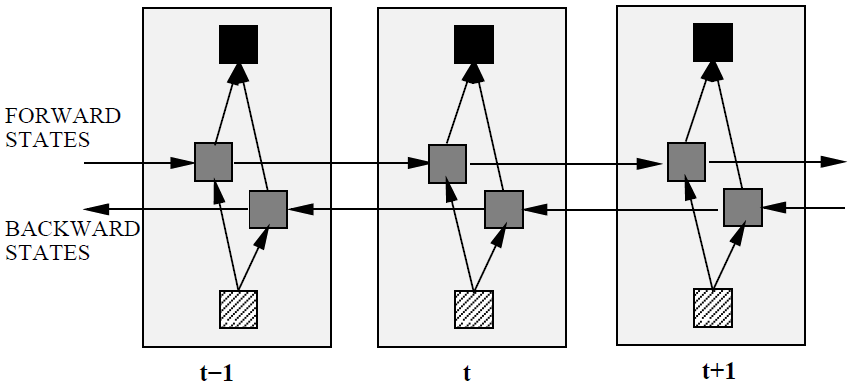


شکل ۳-۹) ساختار بلوک حافظه با سه دروازه [۹۴]

۳-۳-۲-۴- شبکه عصبی بازگشتی دوطرفه

در شبکه‌های عصبی بازگشتی دوطرفه [۸۶, ۹۶]، هر دنباله ورودی در دو جهت زمانی رو به جلو[۱] و از انتها[۲] به دو لایه پنهان بازگشتی کاملا مجزا به نام‌های لایه پیش‌رو[۳] و لایه رو به عقب[۴] داده می‌شود. به‌طوریکه، بین این دو لایه بازگشتی هیچ اتصالی وجود ندارد و هر دو لایه پنهان به لایه خروجی متصل شده‌اند.

مزیت عمده این شبکه‌ها نسبت به شبکه‌های یک‌طرفه[۵] این است که، برخلاف شبکه‌های عصبی یک‌طرفه که مقدار خروجی در هر گام زمانی تنها به مقادیر ورودی قبلی و مقدار ورودی فعلی بستگی دارد، در این شبکه‌ها مقدار خروجی در هر گام زمانی به کل دنباله ورودی بستگی دارد. شکل ۳-۱۰ ساختار یک شبکه عصبی دوطرفه را نشان می‌دهد. اگر نرون‌های لایه پنهان را بلوک‌های حافظه جایگزین نماییم شبکه حافظه کوتاه مدت ماندگار دوطرفه [۱۲, ۹۴] حاصل خواهد شد.



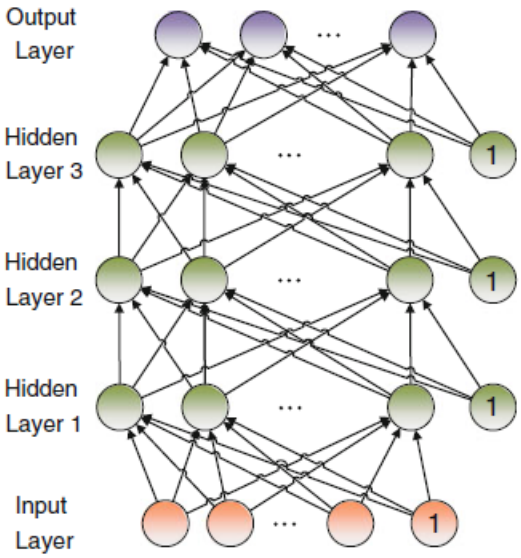
شکل ۳-۱۰) ساختار کلی شبکه‌های عصبی بازگشتی دوطرفه [۹۶]

۳-۳-۲-۵- شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه

یک شبکه عصبی عمیق در حالت اصلی، یک شبکه MLP متداول است که با روی هم قرار دادن تعدادی لایه پنهان (معمولا تعداد این لایه‌ها بیش از دولایه می‌باشد) حاصل می‌شود [۱, ۵]. در این شبکه‌ها خروجی یک لایه پنهان، ورودی

[۱] Forward
[۲] Backward
[۳] Forward Layer
[۴] Backward Layer
[۵] Unidirectional Neural Networks

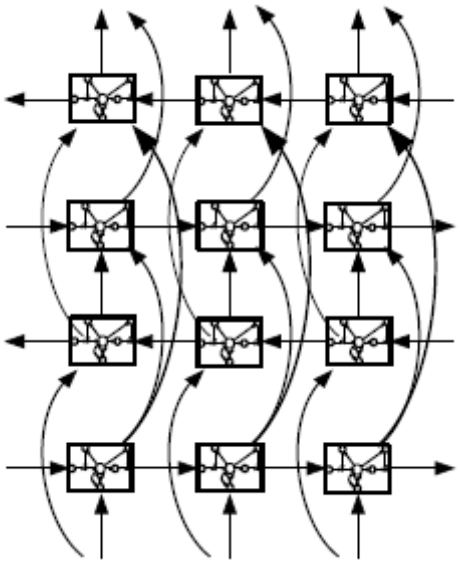
لایه بعدی می‌باشد. در ابتدا مفهوم شبکه عصبی عمیق[۱] به شبکه MLP با بیش از یک لایه پنهان گفته شد. اما بعدها این مفهوم گسترش یافت و به هر شبکه عصبی که شامل تعدادی لایه پنهان روی هم قرار داده شده باشد نسبت داده شد. بنابراین اگر نرون‌های لایه‌های پنهان را با بلوک حافظه LSTM جایگزین نماییم، شبکه حاصل شبکه عصبی حافظه کوتاه مدت ماندگار عمیق خواهد بود. شکل ۳-۱۱ ساختار یک شبکه‌ی عصبی عمیق پنج لایه که شامل سه لایه پنهان می‌باشد را نمایش می‌دهد.



شکل ۳-۱۱) ساختار شبکه‌ی عصبی عمیق [۱]

اگر هر لایه پنهان شبکه عصبی عمیق را با لایه پنهان شبکه عصبی دوطرفه جایگزین کنیم، شبکه عصبی عمیق دوطرفه حاصل می‌گردد. برای این منظور باید هر لایه پنهان شبکه عصبی عمیق با لایه پیش‌رو و رو به عقب شبکه دوطرفه جایگزین گردد و هر لایه پنهان از هر دو لایه پیش‌رو و رو به عقب سطح پایین‌تر شبکه ورودی بگیرد. به‌صورت مشابه اگر در ساختار شبکه عصبی عمیق دوطرفه تمامی نرون‌های لایه‌های پنهان را با بلوک حافظه LSTM را جایگزین شود، شبکه حاصل شبکه عصبی عمیق دوطرفه حافظه کوتاه مدت ماندگار حاصل خواهد شد. ساختار شبکه عصبی عمیق دوطرفه حافظه کوتاه مدت ماندگار در شکل ۳-۱۲ نمایش داده‌شده است.

[۱] Deep Neural Network

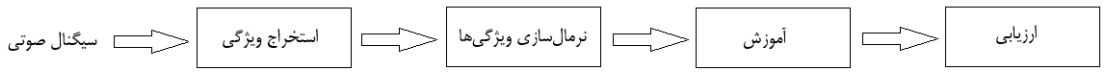


شکل ۳-۱۲) ساختار شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه [۵]

فصل چهارم: روش پیشنهادی– بازشناسی گفتار با شبکه‌ی عمیق

۴-۱- مقدمه

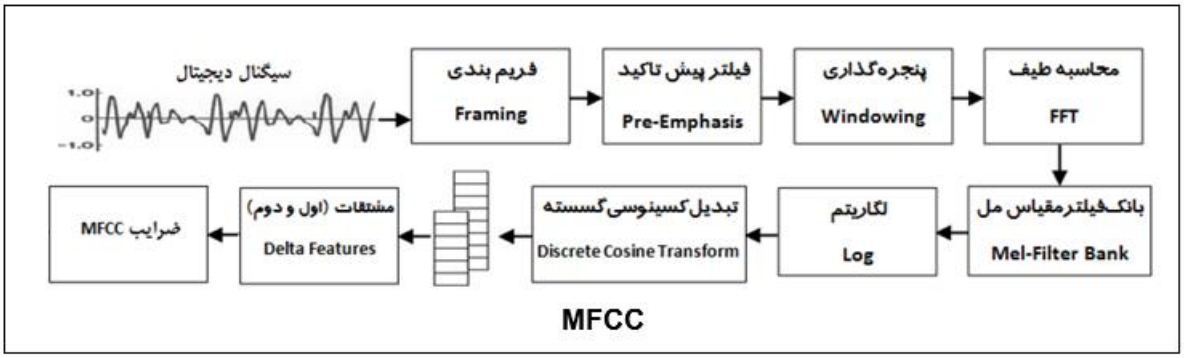
در این فصل مراحل اجرای پایان‌نامه شرح داده خواهد شد. شکل ۴-۱ مراحل اجرای پایان‌نامه را نمایش می‌دهد. به‌منظور طراحی سیستم تشخیص گفتار ابتدا باید هر سیگنال صوتی را تبدیل به تعدادی فریم کنیم و از هرفریم تعدادی ویژگی استخراج نماییم. پس از آن به‌منظور افزایش دقت و کارایی سیستم ویژگی‌های استخراج شده را نرمال‌سازی می‌کنیم. در مرحله بعد با مجموعه داده‌های آموزش شبکه عصبی را آموزش می‌دهیم و پارامترهای مدل را استخراج می‌نماییم. در انتها به کمک مدل به‌دست آمده شبکه را با داده‌های تست ارزیابی می‌کنیم. در ادامه کلیه مراحل شرح داده خواهد شد.



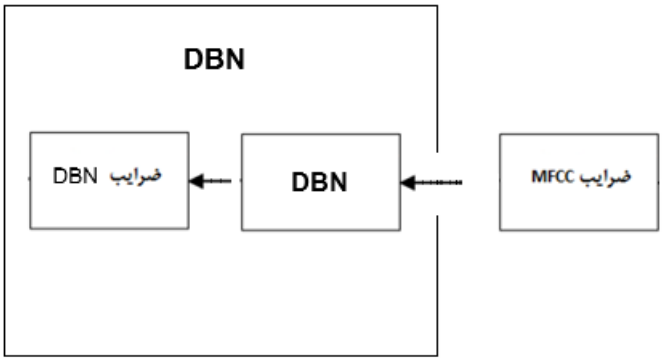
شکل ۴-۱) مراحل اجرای پایان‌نامه

۴-۲- استخراج ویژگی

در این بخش روش‌های مورد استفاده جهت استخراج ویژگی که شامل دو روش MFCC [۱۶] و DBN [۱۷] می‌باشد را شرح می‌دهیم. شکل‌های ۴-۲ و ۴-۳ به‌ترتیب مراحل استخراج ویژگی با MFCC و DBN را نمایش می‌دهند. همان‌طور که در شکل ۴-۳ دیده می‌شود، داده ورودی در روش استخراج ویژگی با استفاده از DBN، ویژگی‌های استخراج شده از روش MFCC می‌باشد. در ادامه این روش‌ها را شرح داده خواهد شد.



شکل ۴-۲) مراحل استخراج ویژگی‌های MFCC



شکل ۴-۳) مراحل استخراج ویژگی‌های DBN

۴-۲-۱- استخراج ویژگی با ضرایب کپسترال در مقیاس مل

ابتدا هر سیگنال به تعدادی فریم تبدیل می‌گردد. برای این‌که هر فریم از نظر آماری ایستا باشد، معمولا طول هر فریم را بین ۲۰ تا ۳۰ میلی ثانیه در نظر گرفته می‌شود. در مرحله بعد به‌منظور حذف اثرات طیفی حنجره و لب‌ها و

همچنین تقویت فرکانس‌های بالای سیگنال، روی فریم‌ها یک فیلتر پیش‌تاکید[۱] اعمال می‌شود. در گام سوم فریم‌ها پنجره‌گذاری[۲] می‌شوند که به‌طور معمول پنجره مورد استفاده پنجره همینگ[۳] است. سپس در گام چهارم با استفاده از تبدیل فوریه سریع[۴] (FFT) توان طیف محاسبه می‌گردد. در مرحله بعد طیف به‌دست آمده را از بانک فیلتر در مقیاس مل[۵] عبور داده می‌شود. مقیاس مل در واقع حساسیت گوش انسان به فرکانس‌های مختلف را مدل می‌کند. با توجه به این‌که گوش انسان به فرکانس‌های پایین اهمیت بیشتری می‌دهد، بنابراین نگاشت این مقیاس برای فرکانس‌های کمتر از ۱۰۰۰ هرتز خطی و برای فرکانس‌های بالاتر به‌صورت لگاریتمی می‌باشد. پس از اعمال این نگاشت معمولا تعداد ۲۴ فیلتر با همپوشانی ۵۰٪ و پهنای یکسان روی فریم‌ها اعمال می‌گردد و مقدار انرژی زیر هر فیلتر محاسبه می‌شود. در گام ششم از انرژی‌های به‌دست آمده لگاریتم گرفته می‌شود و پس از آن در گام هفتم از لگاریتم اندازه انرژی زیر هر فیلتر تبدیل کسینوسی گسسته[۶] (DCT) گرفته می‌شود که اعداد به‌دست آمده در این مرحله ضرایب MFCC هستند. معمولا مشتقات اول و دوم ضرایب MFCC نیز به بردار ویژگی[۷] اضافه می‌گردد.

۴-۲-۲- استخراج ویژگی با استفاده از شبکه باور عمیق

به‌منظور استخراج ویژگی با استفاده از شبکه DBN، از شبکه باور عمیق خود رمزگذار[۸] استفاده می‌کنیم [۹۷]. شبکه باور عمیق خود رمزگذار شامل دو شبکه DBN رمزگذار[۹] و رمزگشا[۱۰] می‌باشد. بخش رمزگذار آن جهت کاهش ابعاد داده ورودی است و بخش رمزگشا جهت ساخت دوباره داده ورودی از روی داده‌ی کد شده می‌باشد. هر شبکه DBN از روی هم قرار دادن تعدادی ماشین بولتزمن محدود[۱۱] (RBM) [۹۸] حاصل می‌شود.

در گام نخست باید شبکه باور عمیق خود رمزگذار را آموزش دهیم. برای این منظور، وزن‌های اولیه هر دو بخش رمزگشا و رمزگذار را یکسان و برابر وزن‌های به‌دست آمده از آموزش شبکه DBN قرار می‌دهیم. سپس جهت اصلاح وزن‌های شبکه، ابتدا داده‌های ورودی (ویژگی‌های MFCC) در جهت رو به جلو به شبکه داده می‌شوند. سپس خطا در لایه خروجی محاسبه شده و پس انتشار [۹۰] می‌یابد. پس از آموزش شبکه جهت استخراج ویژگی، داده‌ها به بخش رمزگذار

---

۱ Pre-Emphasis Filter

۲ Windowing

۳ Hamming

۴ Fast Fourier Transform (FFT)

۵ Mel-Filter Bank

۶ Discrete Cosine Transform (DCT)

۷ Feature Vector

۸ DBN Auto-Encoder

۹ Encoder

۱۰ Decoder

۱۱ Restricted Boltzman Machine (RBM)

شبکه داده می‌شود و خروجی بخش رمزگذار ویژگی‌های مورد نظر می‌باشند. در بخش ۷-۴ شبکه باور عمیق و نحوه استفاده از آن توضیح داده شده است.

۴-۳- نرمال‌سازی دادگان

با توجه با این‌که روش MFCC به نویز حساس می‌باشد، به‌منظور افزایش کارایی و مقاوم‌سازی سیستم، هر یک از مولفه‌های بردارهای ویژگی نرمال شده‌اند یعنی هر مولفه بردار ویژگی به برداری با میانگین صفر و انحراف معیار یک تبدیل گردیده است. این روش یکی از ساده‌ترین و موثرترین روش‌ها برای حذف نویز کانال می‌باشد. برای این منظور گام‌های زیر روی بردارهای ویژگی اعمال کرده و از بردارهای جدید به‌عنوان ورودی شبکه استفاده می‌کنیم. در روابط زیر، $|S|$، x و x_i به‌ترتیب تعداد کل سیگنال‌ها، یک سیگنال از مجموعه S و مولفه i ام سیگنال x را نمایش می‌دهند.

۱. ابتدا به کمک رابطه (۱) میانگین بردار را محاسبه کنید.

$$m_i = \frac{1}{|S|}\sum_{x\epsilon S} x_i \qquad (۱)$$

۲. سپس انحراف معیار[۱] را به کمک رابطه (۲) محاسبه کنید.

$$\sigma_i = \sqrt{\frac{1}{|S|}\sum_{x\epsilon S}(x_i - m_i)^2} \qquad (۲)$$

۳. بردار ویژگی استاندارد شده $\hat{x}_i$ را به کمک رابطه (۳) محاسبه نمایید.

$$\hat{x}_i = \frac{(x_i - m_i)}{\sigma_i} \qquad (۳)$$

۴. از بردارهای ویژگی جدید که هر کدام میانگین صفر و انحراف معیار یک دارند به عنوان ورودی شبکه استفاده کنید.

لازم به ذکر است که بردارهای ویژگی هر یک از مجموعه‌های آموزش، آزمون و ارزیابی باید به‌صورت جداگانه نرمال-سازی شوند.

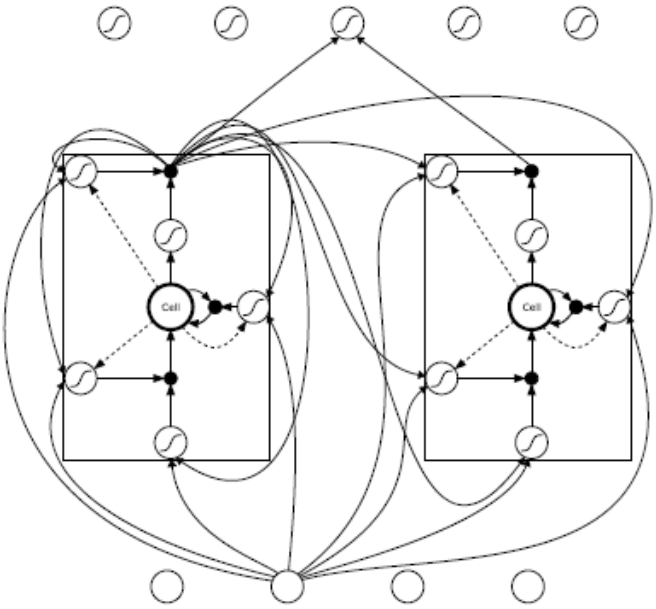
[۱] Standard Deviation

۴-۴- شبکه‌های عصبی حافظه کوتاه مدت ماندگار

در این بخش ساختار، الگوریتم آموزش و آزمون هر یک از شبکه‌های عصبی حافظه کوتاه مدت ماندگار را که در این پژوهش مورد استفاده قرار گرفته است، بیان می‌کنیم.

۴-۴-۱- شبکه عصبی حافظه کوتاه مدت ماندگار یک‌طرفه

با معرفی شبکه LSTM توسط هاکریتر و اشمیدبر در سال ۱۹۹۷ مشکل فراموشی دنباله‌های طولانی در شبکه‌های بازگشتی برطرف گردید [۱۰]. در این شبکه نرون‌های لایه پنهان با بلوک‌های حافظه جایگزین شده‌اند که در ساختار ارائه شده هر بلوک حافظه شامل دو دروازه ورودی و خروجی است. پس از آن در سال ۲۰۰۱ این شبکه توسط گرز توسعه داده شد [۹] و به ساختار بلوک حافطه دروازه فراموشی نیز اضافه شد. شکل۴-۴ ساختار کلی این شبکه را نمایش می‌دهد.

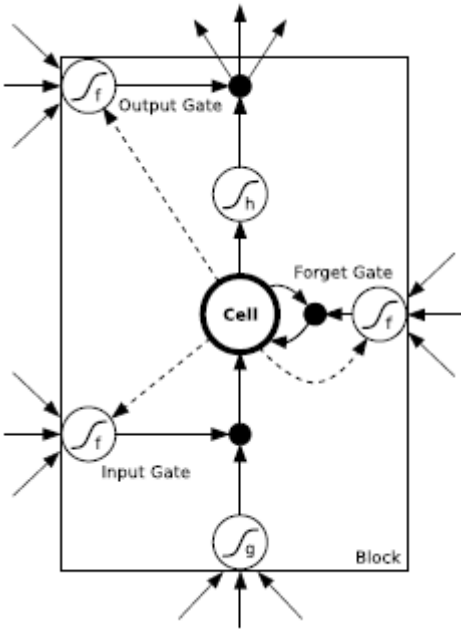


شکل ۴-۴) ساختار شبکه LSTM [۹۴]

۴-۴-۱-۱- ساختار بلوک حافظه

شکل ۴-۵ ساختار بلوک حافظه را نمایش می‌دهد. همان‌طور که در شکل دیده می‌شود، هر بلوک حافظه شامل سه دروازه ورودی، فراموشی، خروجی و همچنین سلول حافظه و تعدادی اتصال از سلول حافظه به دروازه‌ها با نام اتصالات

پیفل[۱] می‌باشد. هر سلول حافظه در مرکز خود یک واحد دارد که به فعال‌ساز آن، حالت[۲] سلول گفته می‌شود. مقدار فعال‌سازی این سه دوازه عددی بین صفر و یک می‌باشد که عدد صفر به معنای بسته بودن کامل دروازه و عدد یک به معنای باز بودن کامل دروازه می‌باشد. در صورتی که دروازه ورودی باز باشد داده ورودی اجازه ورود به لایه پنهان را دارد و در سلول حافظه ذخیره می‌گردد. در غیر این صورت داده امکان ورود به لایه پنهان را ندارد و مقدار آن در سلول حافظه ذخیره نمی‌گردد. اگر دروازه فراموشی باز باشد حالت فعلی سلول تابعی از ورودی جدید و مقادیر داده‌های قبلی خواهد بود و اگر این دروازه بسته باشد حالت سلول تنها به ورودی فعلی وابسته خواهد بود. همچنین در صورت باز بودن دروازه خروجی داده این اجازه را دارد که از لایه پنهان به سمت لایه خروجی حرکت کند. در ادامه الگوریتم‌های آموزش و آزمون این شبکه شرح داده خواهد شد.



شکل ۴-۵) ساختار بلوک حافظه LSTM [۹۴]

۴-۴-۱-۲- الگوریتم آموزش

الگوریتم آموزش شامل دو مرحله پیش‌رو و پس‌رو می‌باشد که در ادامه توضیح داده می‌شوند. نمادهای به‌کار رفته در مراحل آموزش در جدول ۴-۱ ذکر شده است.

[۱] Pheephole Connections

[۲] State

جدول ۴-۱) نمادهای به‌کار رفته در الگوریتم آموزش شبکه LSTM

تعریف	نماد
تعداد نرون‌های لایه پنهان	H
خالص ورودی به سلول در گام زمانی t	a_c^t
خالص ورودی به دروازه ورودی در گام زمانی t	a_ι^t
خالص ورودی به دروازه فراموشی در گام زمانی t	a_ϕ^t
خالص ورودی به دروازه خروجی در گام زمانی t	a_ω^t
حالت سلول در گام زمانی t	s_c^t
فعال‌ساز دروازه در گام زمانی t	b_ι^t
فعال‌ساز دروازه خروجی در گام زمانی t	b_ω^t
فعال‌ساز دروازه فراموشی در گام زمانی t	b_ϕ^t
خروجی سلول در گام زمانی t	b_c^t
خالص ورودی به نرون خروجی k در گام زمانی t	a_k^t
فعال‌ساز نرون خروجی k در گام زمانی t	b_k^t
یال اتصالی از واحد i به واحد j	w_{ij}
دلتای دروازه خروجی در گام زمانی t	δ_ω^t
دلتای نرون خروجی k در گام زمانی t	δ_k^t
دلتای دروازه ورودی در گام زمانی t	δ_ι^t
دلتای دروازه فراموشی در گام زمانی t	δ_ϕ^t
خطای حالت سلول	ϵ_s^t
خطای سلول	ϵ_c^t
فعال‌ساز $h(x)$	$h(x) = \frac{2}{1+e^{-x}} - 1$
فعال‌ساز $f(x)$	$f(x) = \frac{1}{1+e^{-x}}$
فعال‌ساز g(x)	$g(x) = \frac{4}{1+e^{-x}} - 2$
نرخ یادگیری	$0 \leq \alpha \leq 1$

مرحله پیش‌رو

فرض کنید مجموعه آموزش شامل تعدادی دنباله باشد به‌طوریکه، هر دنباله‌ی ورودی X^T و دنباله هدف متناظر آن یعنی Z^T، به فرم زیر باشد:

$$X^T = \{x_1. x_2. \dots . x_{T-1}. x_T\} \quad (۴)$$

$$Z^T = \{z_1. z_2. \dots . z_{T-1}. z_T\} \quad (۵)$$

۱. برای گام زمانی $t = 0$ حالت کلیه سلول‌ها را برابر صفر قرار دهید.

۲. برای گام‌های زمانی $t = 1$ تا $t = T$ گام‌های (۲-۱) تا (۲-۷) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۲-۱. مقدارخالص ورودی به دروازه‌های ورودی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota}\, x_i^t + \sum_{h=1}^{H} w_{h\iota}\, b_h^{t-1} + w_{c\iota}\, s_c^{t-1} \quad (۶)$$

$$b_\iota^t = f(a_\iota^t) \quad (۷)$$

۲-۲. مقدارخالص ورودی به دروازه‌های فراموشی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi}\, x_i^t + \sum_{h=1}^{H} w_{h\phi}\, b_h^{t-1} + w_{c\phi} s_c^{t-1} \quad (۸)$$

$$b_\phi^t = f(a_\phi^t) \quad (۹)$$

۲-۳. مقدار خالص ورودی به سلول‌ها را محاسبه کنید.

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} \quad (۱۰)$$

۲-۴. حالت سلول‌ها را محاسبه کنید.

$$s_c^t = b_\phi^t\, s_c^{t-1} + b_\iota^t\, g(a_c^t) \quad (۱۱)$$

۲-۵. مقدار خالص ورودی به دروازه‌های خروجی، فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega}\, x_i^t + \sum_{h=1}^{H} w_{h\omega}\, b_h^{t-1} + w_{c\omega}\, s_c^t \quad (۱۲)$$

$$b_\omega^t = f(a_\omega^t) \quad (۱۳)$$

۲-۶. خروجی سلول‌ها را محاسبه کنید.

$$b_c^t = b_\omega^t \, h(s_c^t) \quad (۱۴)$$

۲-۷. مقدار خالص ورودی به نرون‌های لایه خروجی و فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_k^t = \sum_{h=1}^{H} w_{hk} \, b_h^t \quad (۱۵)$$

$$b_k^t = f_{softmax}(a_k^t) \quad (۱۶)$$

مرحله پس‌رو

۱. برای گام زمانی $t = T + 1$ تمامی دلتاها را برابر صفر قرار دهید.

۲. برای گام‌های زمانی $t = T$ تا $t = 1$ گام‌های (۲-۱) تا (۲-۱۵) را به‌ترتیب نوشته شده اجرا و مقادیر به‌دست آمده را ذخیره کنید.

۲-۱. دلتای نرون‌های لایه خروجی را محاسبه کنید.

$$\delta_k^t = b_k^t - \, z_k^t \quad (۱۷)$$

۲-۲. خطای سلول‌ها را محاسبه کنید.

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck} \, \delta_k^t + \sum_{g=1}^{G} w_{cg} \, \delta_g^{t+1} \quad (۱۸)$$

۲-۳. دلتای دروازه‌های خروجی را محاسبه کنید.

$$\delta_\omega^t = f'(a_\omega^t) \, h(s_c^t) \, \epsilon_c^t \quad (۱۹)$$

۲-۴. خطای حالت سلول‌ها را به کمک رابطه زیر محاسبه کنید.

$$\epsilon_s^t = b_\omega^t \, h'(s_c^t) \, \epsilon_c^t + b_\phi^{t+1} \, \epsilon_s^{t+1} + w_{c\iota} \, \delta_\iota^{t+1} + w_{c\phi} \, \delta_\phi^{t+1} + w_{c\omega} \, \delta_\omega^t \quad (۲۰)$$

۲-۵. دلتای سلول‌ها را محاسبه کنید.

$$\delta_c^t = b_\iota^t \, g'(a_c^t) \, \epsilon_s^t \quad (۲۱)$$

۲-۶. دلتای دروازه‌های فراموشی را محاسبه کنید.

$$\delta_\phi^t = \, f'\left(a_\phi^t\right) s_c^{t-1} \, \epsilon_s^t \quad (۲۲)$$

۲-۷. دلتای دروازه‌های ورودی را محاسبه کنید.

$$\delta_\iota^t = \, f'(a_\iota^t) \, g(a_c^t) \, \epsilon_s^t \quad (۲۳)$$

۸-۲ تغییرات وزنی یال‌های بین لایه پنهان و لایه خروجی را محاسبه کنید.

$$\Delta w_{hk}^{t} = -\alpha\ \delta_{k}^{t}\ b_{k}^{t} \quad (۲۴)$$

۹-۲ تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های ورودی را محاسبه کنید.

$$\Delta w_{i\iota}^{t} = -\alpha\ x_{i}^{t}\ \delta_{\iota}^{t} \quad (۲۵)$$

۱۰-۲. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های فراموشی را محاسبه کنید.

$$\Delta w_{i\phi}^{t} = -\alpha\ x_{i}^{t}\ \delta_{\phi}^{t} \quad (۲۶)$$

۱۱-۲. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های خروجی را محاسبه کنید.

$$\Delta w_{i\omega}^{t} = -\alpha\ x_{i}^{t}\ \delta_{\omega}^{t} \quad (۲۷)$$

۱۲-۲. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های ورودی را محاسبه کنید.

$$\Delta w_{h\iota}^{t} = -\alpha\ b_{h}^{t}\ \delta_{\iota}^{t} \quad (۲۸)$$

۱۳-۲. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های فراموشی را محاسبه کنید.

$$\Delta w_{h\phi}^{t} = -\alpha\ b_{h}^{t}\ \delta_{\phi}^{t} \quad (۲۹)$$

۱۴-۲. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های خروجی را محاسبه کنید.

$$\Delta w_{h\omega}^{t} = -\alpha\ b_{h}^{t}\ \delta_{\omega}^{t} \quad (۳۰)$$

۱۵-۲. تغییرات وزن اتصالات peephole را محاسبه کنید.

$$\Delta w_{cm}^{t} = -\alpha\ s_{c}^{t}\ \delta_{m}^{t}\ ;\ m \in \{\omega.\phi.\iota\} \quad (۳۱)$$

۳. تمامی وزن‌ها را به کمک رابطه زیر بروز رسانی نمایید.

$$w_{ij} = w_{ij} + \sum_{t=1}^{T} \Delta w_{ij}^{t} \quad (۳۲)$$

۴-۴-۱-۳- ارزیابی شبکه

فرض کنید مجموعه تست شامل تعدادی دنباله باشد بطوریکه هر دنباله‌ی ورودی X^T و دنباله هدف متناظر آن یعنی Z^T، به فرم زیر باشد:

$$X^{T} = \{x_1.x_2.\ldots.x_{T-1}.x_T\} \quad (۳۳)$$

$$Z^{T} = \{z_1.z_2.\ldots.z_{T-1}.z_T\} \quad (۳۴)$$

۱. برای گام زمانی $t = 0$ حالت کلیه سلول‌ها را برابر صفر قرار دهید.

۲. برای گام‌های زمانی $t = 1$ تا $t = T$ گام‌های (۲-۱) تا (۲-۸) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۲-۱. مقدارخالص ورودی به دروازه‌های ورودی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota}\, x_i^t + \sum_{h=1}^{H} w_{h\iota}\, b_h^{t-1} + w_{c\iota}\, s_c^{t-1} \quad (۳۵)$$

$$b_\iota^t = f(a_\iota^t) \quad (۳۶)$$

۲-۲. مقدارخالص ورودی به دروازه‌های فراموشی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi}\, x_i^t + \sum_{h=1}^{H} w_{h\phi}\, b_h^{t-1} + w_{c\phi} s_c^{t-1} \quad (۳۷)$$

$$b_\phi^t = f(a_\phi^t) \quad (۳۸)$$

۲-۳. مقدار خالص ورودی به سلول‌ها را محاسبه کنید.

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} \quad (۳۹)$$

۲-۴. حالت سلول‌ها را محاسبه کنید.

$$s_c^t = b_\phi^t\, s_c^{t-1} + b_\iota^t\, g(a_c^t) \quad (۴۰)$$

۲-۵. مقدار خالص ورودی به دروازه‌های خروجی، فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega}\, x_i^t + \sum_{h=1}^{H} w_{h\omega}\, b_h^{t-1} + w_{c\omega}\, s_c^t \quad (۴۱)$$

$$b_\omega^t = f(a_\omega^t) \quad (۴۲)$$

۲-۶ خروجی سلول‌ها را محاسبه کنید.

$$b_c^t = b_\omega^t\, h(s_c^t) \quad (۴۳)$$

۲-۷. مقدار خالص ورودی به نرون‌های لایه خروجی و فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_k^t = \sum_{h=1}^{H} w_{hk}\, b_h^t \quad (۴۴)$$

$$b_k^t = f_{softmax}(a_k^t) \quad (۴۵)$$

۲-۸. نرون خروجی با مقدار بیشینه را به عنوان خروجی شبکه در گام زمانی t در نظر بگیرید.

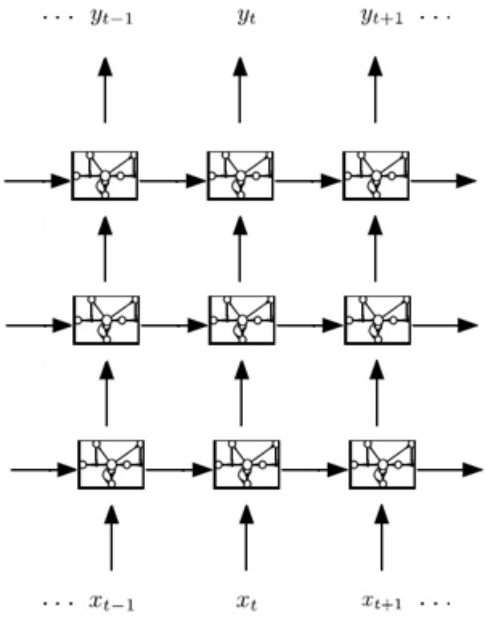
۴-۴-۲- شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه

در این بخش به توضیح پیرامون ساختار، آموزش و نحوه ارزیابی شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه می‌پردازیم.

علت نام‌گذاری یک‌طرفه در این شبکه این است که دنباله ورودی تنها در یک جهت یعنی از اولین گام زمانی تا آخرین گام زمانی به‌ترتیب به لایه‌های پنهان بازگشتی داده می‌شود. درحالی‌که در شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه داده وروی در دو جهت زمانی کاملا مخالف به لایه‌های بازگشتی پیش‌رو و رو به عقب که از یکدیگر مجزا هستند، داده می‌شود. این بدان معنی است که دنباله ورودی به‌ترتیب از گام زمانی اول تا آخرین گام زمانی به لایه‌های بازگشتی پیش‌رو و از گام زمانی آخر تا گام زمانی اول به لایه‌های بازگشتی رو به عقب داده می‌شود که این امر سبب می‌گردد، خروجی شبکه در هر گام زمانی به کل دنباله وابسته باشد. در حالی‌که در شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه خروجی شبکه در هر گام زمانی تنها به اولین گام زمانی تا گام زمانی فعلی وابسته است.

۴-۴-۲-۱- ساختار

شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه از روی هم قرار دادن چندین لایه پنهان که در آن نرون‌های لایه‌های پنهان با بلوک حافظه LSTM جایگزین شده‌اندم، حاصل می‌گردد. شکل ۴-۶ ساختار شبکه عصبی عمیق حافظه کوتاه مدت ماندگار با سه لایه پنهان را نمایش می‌دهد. همان‌طور که در این شکل دیده می‌شود، خروجی هر لایه پنهان ورودی لایه پنهان بالاتر است.



شکل ۴-۶) ساختار شبکه DLSTM

۴-۴-۲-۲- آموزش شبکه

آموزش شبکه شامل دو مرحله پیش‌رو و پس‌رو می‌باشد. در ادامه هر یک از این گام‌ها توضیح داده خواهد شد.

مرحله پیش‌رو

فرض کنید مجموعه آموزش شامل تعدادی دنباله باشد بطوریکه هر دنباله‌ی ورودی X^T و دنباله هدف متناظر آن یعنی Z^T، به فرم زیر باشد:

$$X^T = \{x_1. x_2. \dots . x_{T-1}. x_T\} \quad (۴۶)$$

$$Z^T = \{z_1. z_2. \dots . z_{T-1}. z_T\} \quad (۴۷)$$

در مرحله پیش‌رو آموزش شبکه، ابتدا فعال‌سازهای مربوط به نرون‌های لایه پنهان اول محاسبه می‌گردد. سپس فعال‌سازهای این لایه به عنوان ورودی نرون‌های لایه پنهان دوم در نظر گرفته می‌شود و این روند برای لایه‌های بالاتر تکرار می‌شود. اگر تعداد لایه‌های پنهان برابر L باشد، برای اجرای مرحله پیش‌رو گام‌های ۱ و ۲ را به‌ترتیب مشخص شده اجرا نمایید.

۱. برای لایه پنهان $l = 1$ تا $l = L$ گام‌های (۱-۱) و (۱-۲) را به‌ترتیب اجرا کنید.

۱-۱. برای گام زمانی $t = 0$ حالت کلیه سلول‌ها را برابر صفر قرار دهید.

۱-۲. برای گام‌های زمانی $t = 1$ تا $t = T$ گام‌های (۱-۲-۱) تا (۱-۲-۶) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۱-۲-۱. مقدارخالص ورودی به دروازه‌های ورودی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota}\, x_i^t + \sum_{h=1}^{H} w_{h\iota}\, b_h^{t-1} + w_{c\iota}\, s_c^{t-1} \quad (۴۸)$$

$$b_\iota^t = f(a_\iota^t) \quad (۴۹)$$

۱-۲-۲. مقدارخالص ورودی به دروازه‌های فراموشی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi}\, x_i^t + \sum_{h=1}^{H} w_{h\phi}\, b_h^{t-1} + w_{c\phi} s_c^{t-1} \quad (۵۰)$$

$$b_\phi^t = f(a_\phi^t) \quad (۵۱)$$

۱-۲-۳. مقدار خالص ورودی به سلول‌ها را محاسبه کنید.

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} \quad (۵۲)$$

۱-۲-۴. حالت سلول‌ها را محاسبه کنید.

$$s_c^t = b_\phi^t\, s_c^{t-1} + b_\iota^t\, g(a_c^t) \quad (۵۳)$$

۱-۲-۵. مقدار خالص ورودی به دروازه‌های خروجی، فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega}\, x_i^t + \sum_{h=1}^{H} w_{h\omega}\, b_h^{t-1} + w_{c\omega}\, s_c^t \quad (۵۴)$$

$$b_\omega^t = f(a_\omega^t) \quad (۵۵)$$

۱-۲-۶. خروجی سلول‌ها را محاسبه کنید.

$$b_c^t = b_\omega^t\, h(s_c^t) \quad (۵۶)$$

۲. محاسبه فعال‌سازهای لایه خروجی

برای گام‌های زمانی $t = 1$ تا $t = T$ فعال‌ساز نرون‌های لایه خروجی را به‌کمک رابطه‌های (۵۷) و (۵۸) محاسبه نمایید. در این روابط b_L^t به فعال‌سازهای نرون‌های آخرین لایه پنهان در گام زمانی t اشاره می‌کنند. همچنین w_{Lk} وزن بین آخرین لایه پنهان و خروجی اشاره می‌کند.

$$a_k^t = \sum_{h=1}^{H} w_{Lk}\, b_L^t \quad (۵۷)$$

$$b_k^t = f_{softmax}(a_k^t) \quad (۵۸)$$

مرحله پس‌رو

در مرحله پس‌رو از آموزش شبکه، خطای نرون‌های لایه خروجی پس انتشار می‌یابد. برای اجرای مرحله پس‌رو، مراحل زیر را به‌ترتیب مشخص شده اجرا نمایید.

۱. برای گام‌های زمانی $t = 1$ تا $t = T$ دلتای نرون‌های لایه خروجی را محاسبه کنید.

$$\delta_k^t = b_k^t - z_k^t \quad (۵۹)$$

۲. برای لایه پنهان $l = L$ تا $l = 1$ گام‌های (۲-۱) و (۲-۲) را به‌ترتیب اجرا کنید.

۲-۱. برای گام زمانی $t = T + 1$ تمامی دلتاها را برابر صفر قرار دهید.

۲-۲. برای گام‌های زمانی $t = T$ تا $t = 1$ گام‌های (۲-۲-۱) تا (۲-۲-۱۳) را به‌ترتیب نوشته شده اجرا و مقادیر به‌دست آمده را ذخیره کنید.

۲-۲-۱. خطای سلول‌ها را محاسبه کنید.

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\, \delta_k^t + \sum_{g=1}^{G} w_{cg}\, \delta_g^{t+1} \quad (۶۰)$$

۲-۲-۲. دلتای دروازه‌های خروجی را محاسبه کنید.

$$\delta_\omega^t = f'(a_\omega^t)\, h(s_c^t)\, \epsilon_c^t \quad (۶۱)$$

۲-۲-۳. خطای حالت سلول‌ها را به کمک رابطه زیر محاسبه کنید.

$$\epsilon_s^t = b_\omega^t\, h'(s_c^t)\, \epsilon_c^t + b_\phi^{t+1}\, \epsilon_s^{t+1} + w_{c\iota}\, \delta_\iota^{t+1} + w_{c\phi}\, \delta_\phi^{t+1} + w_{c\omega}\, \delta_\omega^t \quad (۶۲)$$

۲-۲-۴. دلتای سلول‌ها را محاسبه کنید.

$$\delta_c^t = b_\iota^t\, g'(a_c^t)\, \epsilon_s^t \quad (۶۳)$$

۲-۲-۵. دلتای دروازه‌های فراموشی را محاسبه کنید.

$$\delta_\phi^t = f'(a_\phi^t)\, s_c^{t-1}\, \epsilon_s^t \quad (۶۴)$$

۲-۲-۶. دلتای دروازه‌های ورودی را محاسبه کنید.

$$\delta_\iota^t = f'(a_\iota^t)\, g(a_c^t)\, \epsilon_s^t \quad (۶۵)$$

۲-۲-۷. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های ورودی را محاسبه کنید.

$$\Delta w_{i\iota}^{t} = -\alpha\, x_i^t\, \delta_\iota^t \quad (۶۶)$$

۲-۲-۸. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های فراموشی را محاسبه کنید.

$$\Delta w_{i\iota}^{t} = -\alpha\, x_i^t\, \delta_\phi^t \quad (۶۷)$$

۲-۲-۹. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های خروجی را محاسبه کنید.

$$\Delta w_{i\omega}^{t} = -\alpha\, x_i^t\, \delta_\omega^t \quad (۶۸)$$

۲-۲-۱۰. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های ورودی را محاسبه کنید.

$$\Delta w_{h\iota}^{t} = -\alpha\, b_h^t\, \delta_\iota^t \quad (۶۹)$$

۲-۲-۱۱. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های فراموشی را محاسبه کنید.

$$\Delta w_{h\phi}^{t} = -\alpha\, b_h^t\, \delta_\phi^t \quad (۷۰)$$

۲-۲-۱۲. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های خروجی را محاسبه کنید.

$$\Delta w_{h\omega}^{t} = -\alpha\, b_h^t\, \delta_\omega^t \quad (۷۱)$$

۲-۲-۱۳. تغییرات وزن اتصالات peephole را محاسبه کنید.

$$\Delta w_{cm}^{t} = -\alpha\, s_c^t\, \delta_m^t\ ;\ m \in \{\omega.\phi.\iota\} \quad (۷۲)$$

۳. برای گام‌های زمانی $t = 1$ تا $t = T$ تغییرات وزن بین نرون‌های لایه خروجی و آخرین لایه پنهان را با استفاده از رابطه (۷۳) محاسبه کنید.

$$\Delta w_{Lk}^{t} = -\,\alpha\, \delta_k^t\, b_L^t \quad (۷۳)$$

۴. تمامی وزن‌ها را با استفاده از رابطه (۷۴) به‌روز رسانی کنید.

$$w_{ij} = w_{ij} + \sum_{t=1}^{T} \Delta w_{ij}^{t} \quad (۷۴)$$

۴-۴-۲-۳- ارزیابی شبکه

فرض کنید مجموعه تست شامل تعدادی دنباله باشد بطوریکه هر دنباله‌ی ورودی X^T و دنباله هدف متناظر آن یعنی Z^T، به فرم زیر باشد:

$$X^T = \{x_1.x_2.\ldots.x_{T-1}.x_T\} \quad (۷۵)$$

$$Z^T = \{z_1.z_2.\ldots.z_{T-1}.z_T\} \quad (۷۶)$$

به‌منظور ارزیابی شبکه، گام‌های ۱ و۲ را به‌ترتیب مشخص شده اجرا نمایید.

۱. برای لایه پنهان $l = 1$ تا $l = L$ گام‌های (۱-۱) و (۱-۲) را به‌ترتیب اجرا کنید.

۱-۱. برای گام زمانی $t = 0$ حالت کلیه سلول‌ها را برابر صفر قرار دهید.

۱-۲. برای گام‌های زمانی $t = 1$ تا $t = T$ گام‌های (۱-۲-۱) تا (۱-۲-۶) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۱-۲-۱. مقدارخالص ورودی به دروازه‌های ورودی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota}\, x_i^t + \sum_{h=1}^{H} w_{h\iota}\, b_h^{t-1} + w_{c\iota}\, s_c^{t-1} \quad (۷۷)$$

$$b_\iota^t = f(a_\iota^t) \quad (۷۸)$$

۱-۲-۲. مقدارخالص ورودی به دروازه‌های فراموشی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi}\, x_i^t + \sum_{h=1}^{H} w_{h\phi}\, b_h^{t-1} + w_{c\phi} s_c^{t-1} \quad (۷۹)$$

$$b_\phi^t = f(a_\phi^t) \quad (۸۰)$$

۱-۲-۳. مقدار خالص ورودی به سلول‌ها را محاسبه کنید.

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} \quad (۸۱)$$

۱-۲-۴. حالت سلول‌ها را محاسبه کنید.

$$s_c^t = b_\phi^t\, s_c^{t-1} + b_\iota^t\, g(a_c^t) \quad (۸۲)$$

۱-۲-۵. مقدار خالص ورودی به دروازه‌های خروجی، فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega}\, x_i^t + \sum_{h=1}^{H} w_{h\omega}\, b_h^{t-1} + w_{c\omega}\, s_c^t \quad (۸۳)$$

$$b_\omega^t = f(a_\omega^t) \quad (۸۴)$$

۱-۲-۶. خروجی سلول‌ها را محاسبه کنید.

$$b_c^t = b_\omega^t\, h(s_c^t) \quad (۸۵)$$

۲. محاسبه فعال‌سازهای لایه خروجی

برای گام‌های زمانی $t = 1$ تا $t = T$ فعال‌ساز نرون‌های لایه خروجی را به‌کمک رابطه‌های (۸۶) و (۸۷) محاسبه نمایید و نرون خروجی با بیشترین مقدار فعال‌سازی را به عنوان خروجی شبکه درنظر بگیرید. در این روابط b_L^t به

فعال‌ساز نرون‌های آخرین لایه پنهان در گام زمانی t اشاره می‌کند. همچنین w_{Lk} وزن بین آخرین لایه پنهان و لایه خروجی می‌باشد.

$$a_k^t = \sum_{h=1}^{H} w_{Lk}\, b_L^t \quad (۸۶)$$

$$b_k^t = f_{softmax}(a_k^t) \quad (۸۷)$$

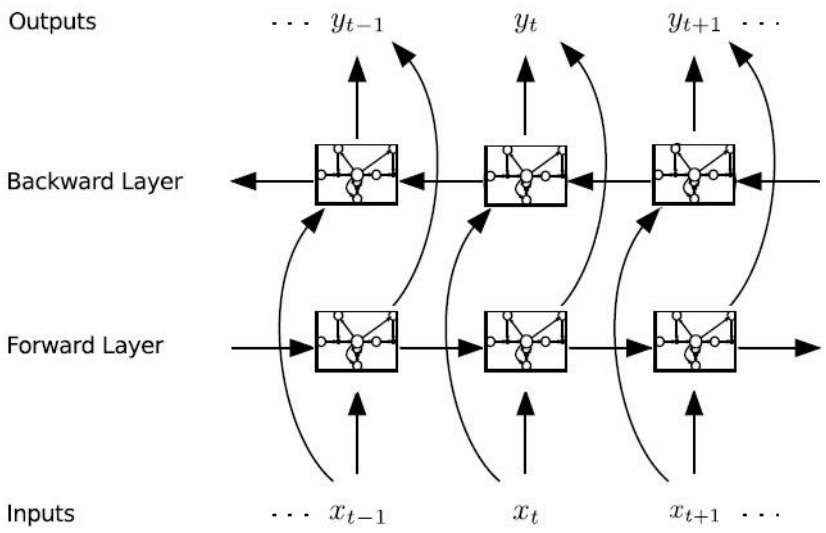
۴-۴-۳- شبکه عصبی حافظه کوتاه مدت ماندگار دوطرفه

در ابتدا به توضیح پیرامون ساختار شبکه عصبی حافظه کوتاه مدت ماندگار دوطرفه می‌پردازیم و در ادامه نحوه آموزش و استفاده از این شبکه توضیح می‌دهیم.

برخلاف شبکه عصبی حافظه کوتاه مدت ماندگار یک‌طرفه که تنها شامل یک لایه پنهان بازگشتی می‌باشد، شبکه عصبی حافظه کوتاه مدت ماندگار دوطرفه [۱۲] شامل دو لایه پنهان بازگشتی مجزا است که بین این دو لایه هیچ اتصالی وجود ندارد و هر دو لایه به لایه خروجی متصل شده‌اند. دنباله ورودی از اولین گام زمانی تا آخرین گام زمانی به‌ترتیب به لایه پیش‌رو داده می‌شود و از آخرین گام زمانی تا اولین گام زمانی به لایه رو به عقب داده می‌شود. بنابراین در حالت دوطرفه مقدار خروجی شبکه در هر لحظه به کل دنباله ورودی وابسته خواهد شد.

۴-۴-۳-۱- ساختار

همان‌طور که در شکل ۴-۷ دیده می‌شود، شبکه عصبی BLSTM شامل دو لایه پنهان بازگشتی مجزا با بلوک‌های حافظه LSTM می‌باشد. بین این دو لایه هیچ اتصالی وجود ندارد و هر دو لایه پنهان به لایه خروجی متصل شده‌اند.



شکل ۴-۷) ساختار شبکه عصبی حافظه کوتاه مدت ماندگار دوطرفه [۶]

۴-۴-۳-۲- آموزش شبکه

آموزش این شبکه همانند آموزش شبکه LSTM یک‌طرفه شامل دو مرحله پیش‌رو و رو به عقب می‌باشد که در ادامه به توضیح آن می‌پردازیم.

مرحله پیش‌رو

فرض کنید مجموعه آموزش شامل تعدادی دنباله باشد بطوریکه هر دنباله‌ی ورودی X^T و دنباله هدف متناظر آن یعنی Z^T، به فرم زیر باشد:

$$X^T = \{x_1. x_2. \dots . x_{T-1}. x_T\} \quad (۸۸)$$

$$Z^T = \{z_1. z_2. \dots . z_{T-1}. z_T\} \quad (۸۹)$$

در مرحله پیش‌رو، داده‌های ورودی از گام زمانی $t = 1$ تا $t = T$ به لایه پیش‌رو داده می‌شوند و در جهت کاملا عکس یعنی از گام زمانی $t = T$ تا $t = 1$ به لایه رو به عقب داده می‌شوند. برای اجرای مرحله پیش‌رو، گام‌های ۱ تا ۳ را به‌ترتیب مشخص شده اجرا نمایید.

۱. مرحله پیش‌رو لایه پیش‌رو

در این مرحله بردارهای ورودی به‌ترتیب از گام زمانی $t = 1$ تا $t = T$ به این لایه داده می‌شوند و فعال‌سازهای مربوط به تمامی گام‌های زمانی ذخیره می‌گردد. برای انجام مرحله پیش‌رو مربوط به لایه پیش‌رو گام‌های (۱-۱) و (۱-۲) را به‌ترتیب مشخص شده انجام دهید.

۱-۱. برای گام زمانی $t = 0$ حالت کلیه سلول‌ها را برابر صفر قرار دهید.

۱-۲. برای گام‌های زمانی $t = 1$ تا $t = T$ گام‌های (۱-۲-۱) تا (۱-۲-۶) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۱-۲-۱. مقدارخالص ورودی به دروازه‌های ورودی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota}\, x_i^t + \sum_{h=1}^{H} w_{h\iota}\, b_h^{t-1} + w_{c\iota}\, s_c^{t-1} \quad (۹۰)$$

$$b_\iota^t = f(a_\iota^t) \quad (۹۱)$$

۱-۲-۲. مقدارخالص ورودی به دروازه‌های فراموشی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi}\, x_i^t + \sum_{h=1}^{H} w_{h\phi}\, b_h^{t-1} + w_{c\phi} s_c^{t-1} \quad (۹۲)$$

$$b_\phi^t = f(a_\phi^t) \quad (۹۳)$$

۱-۲-۳. مقدار خالص ورودی به سلول‌ها را محاسبه کنید.

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} \quad (۹۴)$$

۱-۲-۴. حالت سلول‌ها را محاسبه کنید.

$$s_c^t = b_\phi^t\, s_c^{t-1} + b_\iota^t\, g(a_c^t) \quad (۹۵)$$

۱-۲-۵. مقدار خالص ورودی به دروازه‌های خروجی، فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega}\, x_i^t + \sum_{h=1}^{H} w_{h\omega}\, b_h^{t-1} + w_{c\omega}\, s_c^t \quad (۹۶)$$

$$b_\omega^t = f(a_\omega^t) \quad (۹۷)$$

۱-۲-۶. خروجی سلول‌ها را محاسبه کنید.

$$b_c^t = b_\omega^t\, h(s_c^t) \quad (۹۸)$$

۲. مرحله پیش‌رو لایه رو به عقب

در این مرحله بردارهای ورودی به‌ترتیب از گام زمانی $t = T$ تا $t = 1$ به لایه رو به عقب داده می‌شوند و فعال‌سازهای مربوط به تمامی گام‌های زمانی ذخیره می‌گردد. برای انجام مرحله پیش‌رو مربوط به لایه رو به عقب گام‌های (۲-۱) و (۲-۲) را به‌ترتیب مشخص شده انجام دهید.

۲-۱. برای گام زمانی $t = T + 1$ حالت کلیه سلول‌ها را برابر صفر قرار دهید.

۲-۲. برای گام‌های زمانی $t = T$ تا $t = 1$ گام‌های (۲-۲-۱) تا (۲-۲-۶) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۲-۲-۱. مقدارخالص ورودی به دروازه‌های ورودی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota}\, x_i^t + \sum_{h=1}^{H} w_{h\iota}\, b_h^{t-1} + w_{c\iota}\, s_c^{t-1} \quad (۹۹)$$

$$b_\iota^t = f(a_\iota^t) \quad (۱۰۰)$$

۲-۲-۲. مقدارخالص ورودی به دروازه‌های فراموشی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi}\, x_i^t + \sum_{h=1}^{H} w_{h\phi}\, b_h^{t-1} + w_{c\phi} s_c^{t-1} \quad (۱۰۱)$$

$$b_\phi^t = f(a_\phi^t) \quad (۱۰۲)$$

۲-۲-۳. مقدار خالص ورودی به سلول‌ها را محاسبه کنید.

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} \quad (۱۰۳)$$

۲-۲-۴. حالت سلول‌ها را محاسبه کنید.

$$s_c^t = b_\phi^t\, s_c^{t-1} + b_\iota^t\, g(a_c^t) \quad (۱۰۴)$$

۲-۲-۵. مقدار خالص ورودی به دروازه‌های خروجی، فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega}\, x_i^t + \sum_{h=1}^{H} w_{h\omega}\, b_h^{t-1} + w_{c\omega}\, s_c^t \quad (۱۰۵)$$

$$b_\omega^t = f(a_\omega^t) \quad (۱۰۶)$$

۲-۲-۶. خروجی سلول‌ها را محاسبه کنید.

$$b_c^t = b_\omega^t\, h(s_c^t) \quad (۱۰۷)$$

۳. محاسبه فعال‌سازهای لایه خروجی

برای گام‌های زمانی $t = 1$ تا $t = T$ فعال‌ساز نرون‌های لایه خروجی را به‌کمک رابطه‌های (۱۰۸) و (۱۰۹) محاسبه نمایید. در این روابط b_b^t و b_f^t به‌ترتیب به فعال‌ساز نرون‌های لایه رو به عقب و پیش‌رو در گام زمانی t اشاره

می‌کنند. همچنین w_{bk} و w_{fk} به‌ترتیب به وزن بین لایه رو به عقب و خروجی و وزن بین لایه پیش‌رو و خروجی اشاره می‌کنند.

$$a_k^t = \sum_{h=1}^{H} w_{fk}\, b_f^t + w_{bk}\, b_b^t \quad (۱۰۸)$$

$$b_k^t = f_{softmax}(a_k^t) \quad (۱۰۹)$$

مرحله پس‌رو

در مرحله پس‌رو، خطای نرون‌های لایه خروجی برای لایه پیش‌رو از گام زمانی $t = T$ تا $t = 1$ در لایه پیش‌رو پس انتشار می‌یابد و برای لایه رو به عقب کاملا در جهت عکس و از گام زمانی $t = 1$ تا $t = T$ به لایه رو به عقب پس انتشار می‌یابد.

برای اجرای مرحله پس‌رو، گام‌های ۱ تا ۳ را به‌ترتیب مشخص شده اجرا نمایید.

۱. مرحله پس‌رو لایه پیش‌رو

در این مرحله خطای نرون‌های لایه خروجی به‌ترتیب از گام زمانی $t = T$ تا $t = 1$ در لایه پیش‌رو پس‌انتشار می‌یابد و دلتای تمامی دورازه‌ها و سلول‌ها به کمک مقادیر ذخیره شده در مرحله پیش‌رو این لایه محاسبه می‌گردد. برای انجام مرحله پس‌رو مربوط به لایه پیش‌رو گام‌های (۱-۱) و (۱-۲) را به‌ترتیب مشخص شده انجام دهید.

۱-۱. برای گام زمانی $t = T + 1$ کلیه دلتاها را را برابر صفر قرار دهید.

۱-۲. برای گام‌های زمانی $t = T$ تا $t = 1$ گام‌های (۱-۲-۱) تا (۱-۲-۱۵) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۱-۲-۱. دلتای نرون‌های لایه خروجی را محاسبه کنید.

$$\delta_k^t = b_k^t - z_k^t \quad (۱۱۰)$$

۱-۲-۲. خطای سلول‌ها را محاسبه کنید.

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\, \delta_k^t + \sum_{g=1}^{G} w_{cg}\, \delta_g^{t+1} \quad (۱۱۱)$$

۱-۲-۳. دلتای دروازه‌های خروجی را محاسبه کنید.

$$\delta_{\omega}^{t} = f'(a_{\omega}^{t})\, h(s_{c}^{t})\, \epsilon_{c}^{t} \quad (۱۱۲)$$

۱-۲-۴. خطای حالت سلول‌ها را به کمک رابطه زیر محاسبه کنید.

$$\epsilon_{s}^{t} = b_{\omega}^{t}\, h'(s_{c}^{t})\, \epsilon_{c}^{t} + b_{\phi}^{t+1}\, \epsilon_{s}^{t+1} + w_{c\iota}\, \delta_{\iota}^{t+1} + w_{c\phi}\, \delta_{\phi}^{t+1} + w_{c\omega}\, \delta_{\omega}^{t} \quad (۱۱۳)$$

۱-۲-۵. دلتای سلول‌ها را محاسبه کنید.

$$\delta_{c}^{t} = b_{\iota}^{t}\, g'(a_{c}^{t})\, \epsilon_{s}^{t} \quad (۱۱۴)$$

۱-۲-۶. دلتای دروازه‌های فراموشی را محاسبه کنید.

$$\delta_{\phi}^{t} = f'(a_{\phi}^{t})\, s_{c}^{t-1}\, \epsilon_{s}^{t} \quad (۱۱۵)$$

۱-۲-۷. دلتای دروازه‌های ورودی را محاسبه کنید.

$$\delta_{\iota}^{t} = f'(a_{\iota}^{t})\, g(a_{c}^{t})\, \epsilon_{s}^{t} \quad (۱۱۶)$$

۱-۲-۸. تغییرات وزنی یال‌های بین لایه پنهان ولایه خروجی را محاسبه کنید.

$$\Delta w_{hk}^{t} = -\alpha\, \delta_{k}^{t}\, b_{k}^{t} \quad (۱۱۷)$$

۱-۲-۹. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های ورودی را محاسبه کنید.

$$\Delta w_{i\iota}^{t} = -\alpha\, x_{i}^{t}\, \delta_{\iota}^{t} \quad (۱۱۸)$$

۱-۲-۱۰. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های فراموشی را محاسبه کنید.

$$\Delta w_{i\iota}^{t} = -\alpha\, x_{i}^{t}\, \delta_{\phi}^{t} \quad (۱۱۹)$$

۱-۲-۱۱. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های خروجی را محاسبه کنید.

$$\Delta w_{i\omega}^{t} = -\alpha\, x_{i}^{t}\, \delta_{\omega}^{t} \quad (۱۲۰)$$

۱-۲-۱۲. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های ورودی را محاسبه کنید.

$$\Delta w_{h\iota}^{t} = -\alpha\, b_{h}^{t}\, \delta_{\iota}^{t} \quad (۱۲۱)$$

۱-۲-۱۳. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های فراموشی را محاسبه کنید.

$$\Delta w_{h\phi}^{t} = -\alpha\, b_{h}^{t}\, \delta_{\phi}^{t} \quad (۱۲۲)$$

۱-۲-۱۴. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های خروجی را محاسبه کنید.

$$\Delta w_{h\omega}^{t} = -\alpha\, b_{h}^{t}\, \delta_{\omega}^{t} \quad (۱۲۳)$$

۱-۲-۱۵. تغییرات وزن اتصالات peephole را محاسبه کنید.

$$\Delta w_{cm}^{t} = -\alpha\, s_c^t\, \delta_m^t \; ; \; m \in \{\omega . \phi . \iota\} \qquad (۱۲۴)$$

۲. مرحله پس‌رو لایه رو به عقب

در این مرحله خطای لایه خروجی به‌ترتیب از گام زمانی $t = 1$ تا $t = T$ در لایه رو به عقب پس‌انتشار می‌یابد و دلتای تمامی دورازه‌ها و سلول‌ها به کمک مقادیر ذخیره شده در مرحله پیش‌رو این لایه محاسبه می‌گردد. برای انجام مرحله پس‌رو مربوط به لایه رو به عقب گام‌های (۲-۱) و (۲-۲) را به‌ترتیب مشخص شده انجام دهید.

۲-۱. برای گام زمانی $t = 0$ کلیه دلتاها را را برابر صفر قرار دهید.

۲-۲. برای گام‌های زمانی $t = 1$ تا $t = T$ گام‌های (۲-۲-۱) تا (۲-۲-۱۵) را به ت‌تیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۲-۲-۱. دلتای نرون‌های لایه خروجی را محاسبه کنید.

$$\delta_k^t = b_k^t - z_k^t \qquad (۱۲۵)$$

۲-۲-۲. خطای سلول‌ها را محاسبه کنید.

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\, \delta_k^t + \sum_{g=1}^{G} w_{cg}\, \delta_g^{t+1} \qquad (۱۲۶)$$

۲-۲-۳. دلتای دروازه‌های خروجی را محاسبه کنید.

$$\delta_\omega^t = f'(a_\omega^t)\, h(s_c^t)\, \epsilon_c^t \qquad (۱۲۷)$$

۲-۲-۴. خطای حالت سلول‌ها را به کمک رابطه زیر محاسبه کنید.

$$\epsilon_s^t = b_\omega^t\, h'(s_c^t)\, \epsilon_c^t + b_\phi^{t+1}\, \epsilon_s^{t+1} + w_{c\iota}\, \delta_\iota^{t+1} + w_{c\phi}\, \delta_\phi^{t+1} + w_{c\omega}\, \delta_\omega^t \qquad (۱۲۸)$$

۲-۲-۵. دلتای سلول‌ها را محاسبه کنید.

$$\delta_c^t = b_\iota^t\, g'(a_c^t)\, \epsilon_s^t \qquad (۱۲۹)$$

۲-۲-۶. دلتای دروازه‌های فراموشی را محاسبه کنید.

$$\delta_\phi^t = f'\left(a_\phi^t\right) s_c^{t-1}\, \epsilon_s^t \qquad (۱۳۰)$$

۲-۲-۷. دلتای دروازه‌های ورودی را محاسبه کنید.

$$\delta_\iota^t = f'(a_\iota^t)\, g(a_c^t)\, \epsilon_s^t \qquad (۱۳۱)$$

۲-۲-۸. تغییرات وزنی یال‌های بین لایه پنهان ولایه خروجی را محاسبه کنید.

$$\Delta w_{hk}^t = -\alpha\, \delta_k^t\, b_k^t \qquad (۱۳۲)$$

۹-۲-۲. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های ورودی را محاسبه کنید.

$$\Delta w_{i\iota}^{t} = -\alpha\, x_i^t\, \delta_\iota^t \qquad (۱۳۳)$$

۱۰-۲-۲. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های فراموشی را محاسبه کنید.

$$\Delta w_{i\iota}^{t} = -\alpha\, x_i^t\, \delta_\phi^t \qquad (۱۳۴)$$

۱۱-۲-۲. تغییرات وزنی یال‌های بین لایه ورودی و دروازه‌های خروجی را محاسبه کنید.

$$\Delta w_{i\omega}^{t} = -\alpha\, x_i^t\, \delta_\omega^t \qquad (۱۳۵)$$

۱۲-۲-۲. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های ورودی را محاسبه کنید.

$$\Delta w_{h\iota}^{t} = -\alpha\, b_h^t\, \delta_\iota^t \qquad (۱۳۶)$$

۱۳-۲-۲. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های فراموشی را محاسبه کنید.

$$\Delta w_{h\phi}^{t} = -\alpha\, b_h^t\, \delta_\phi^t \qquad (۱۳۷)$$

۱۴-۲-۲. تغییرات وزنی یال‌های بلوک‌های حافظه و دروازه‌های خروجی را محاسبه کنید.

$$\Delta w_{h\omega}^{t} = -\alpha\, b_h^t\, \delta_\omega^t \qquad (۱۳۸)$$

۱۵-۲-۲. تغییرات وزن اتصالات peephole را محاسبه کنید.

$$\Delta w_{cm}^{t} = -\alpha\, s_c^t\, \delta_m^t \;;\; m \in \{\omega.\phi.\iota\} \qquad (۱۳۹)$$

۳. تمامی وزن‌ها را به کمک رابطه (۱۴۰) به‌روز رسانی نمایید.

$$w_{ij} = w_{ij} + \sum_{t=1}^{T} \Delta w_{ij}^{t} \qquad (۱۴۰)$$

۴-۴-۳-۳- ارزیابی شبکه

فرض کنید مجموعه تست شامل تعدادی دنباله باشد بطوریکه هر دنباله‌ی ورودی X^T و دنباله هدف متناظر آن یعنی Z^T، به فرم زیر باشد:

$$X^T = \{x_1.x_2.\dots.x_{T-1}.x_T\} \qquad (۱۴۱)$$

$$Z^T = \{z_1.z_2.\dots.z_{T-1}.z_T\} \qquad (۱۴۲)$$

به‌منظور ارزیابی شبکه، گام‌های ۱ تا ۳ را به‌ترتیب مشخص شده اجرا نمایید.

۱. محاسبه فعال‌سازهای لایه پیش‌رو

۱-۱. برای گام زمانی $t = 0$ حالت کلیه سلول‌ها را برابر صفر قرار دهید.

۱-۲. برای گام‌های زمانی $t = 1$ تا $t = T$ گام‌های (۱-۲-۱) تا (۱-۲-۶) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۱-۲-۱. مقدارخالص ورودی به دروازه‌های ورودی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota}\, x_i^t + \sum_{h=1}^{H} w_{h\iota}\, b_h^{t-1} + w_{c\iota}\, s_c^{t-1} \quad (۱۴۳)$$

$$b_\iota^t = f(a_\iota^t) \quad (۱۴۴)$$

۱-۲-۲. مقدارخالص ورودی به دروازه‌های فراموشی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi}\, x_i^t + \sum_{h=1}^{H} w_{h\phi}\, b_h^{t-1} + w_{c\phi} s_c^{t-1} \quad (۱۴۵)$$

$$b_\phi^t = f(a_\phi^t) \quad (۱۴۶)$$

۱-۲-۳. مقدار خالص ورودی به سلول‌ها را محاسبه کنید.

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} \quad (۱۴۷)$$

۱-۲-۴. حالت سلول‌ها را محاسبه کنید.

$$s_c^t = b_\phi^t\, s_c^{t-1} + b_\iota^t\, g(a_c^t) \quad (۱۴۸)$$

۱-۲-۵. مقدار خالص ورودی به دروازه‌های خروجی، فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega}\, x_i^t + \sum_{h=1}^{H} w_{h\omega}\, b_h^{t-1} + w_{c\omega}\, s_c^t \quad (۱۴۹)$$

$$b_\omega^t = f(a_\omega^t) \quad (۱۵۰)$$

۱-۲-۶. خروجی سلول‌ها را محاسبه کنید.

$$b_c^t = b_\omega^t\, h(s_c^t) \quad (۱۵۱)$$

۲. محاسبه فعال‌سازهای لایه رو به عقب

۲-۱. برای گام زمانی $t = T + 1$ حالت کلیه سلول‌ها را برابر صفر قرار دهید.

۲-۲. برای گام‌های زمانی $t = T$ تا $t = 1$ گام‌های (۲-۲-۱) تا (۲-۲-۶) را به‌ترتیب نوشته شده اجرا و مقادیر آن‌ها را ذخیره کنید.

۲-۲-۱. مقدارخالص ورودی به دروازه‌های ورودی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota}\, x_i^t + \sum_{h=1}^{H} w_{h\iota}\, b_h^{t-1} + w_{c\iota}\, s_c^{t-1} \quad (۱۵۲)$$

$$b_\iota^t = f(a_\iota^t) \quad (۱۵۳)$$

۲-۲-۲. مقدارخالص ورودی به دروازه‌های فراموشی و همچنین فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi}\, x_i^t + \sum_{h=1}^{H} w_{h\phi}\, b_h^{t-1} + w_{c\phi} s_c^{t-1} \quad (۱۵۴)$$

$$b_\phi^t = f(a_\phi^t) \quad (۱۵۵)$$

۲-۲-۳. مقدار خالص ورودی به سلول‌ها را محاسبه کنید.

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} \quad (۱۵۶)$$

۲-۲-۴. حالت سلول‌ها را محاسبه کنید.

$$s_c^t = b_\phi^t\, s_c^{t-1} + b_\iota^t\, g(a_c^t) \quad (۱۵۷)$$

۲-۲-۵. مقدار خالص ورودی به دروازه‌های خروجی، فعال‌سازهای مربوط به آن‌ها را محاسبه کنید.

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega}\, x_i^t + \sum_{h=1}^{H} w_{h\omega}\, b_h^{t-1} + w_{c\omega}\, s_c^t \quad (۱۵۸)$$

$$b_\omega^t = f(a_\omega^t) \quad (۱۵۹)$$

۲-۲-۶. خروجی سلول‌ها را محاسبه کنید.

$$b_c^t = b_\omega^t\, h(s_c^t) \quad (۱۶۰)$$

۳. محاسبه فعال‌سازهای لایه خروجی

برای گام‌های زمانی $t = 1$ تا $t = T$ فعال‌ساز نرون‌های لایه خروجی را به کمک رابطه‌های (۱۶۱) و (۱۶۲) محاسبه نمایید و نرون خروجی با بیشترین مقدار را به عنوان خروجی شبکه در نظر بگیرید. در این روابط b_b^t و b_f^t به‌ترتیب به فعال‌سازهای نرون‌های آخرین لایه رو به عقب و پیش‌رو در گام زمانی t اشاره می‌کنند. همچنین w_{bk} و w_{fk} به‌ترتیب به وزن بین لایه رو به عقب و خروجی و وزن بین لایه پیش‌رو و خروجی اشاره می‌کند.

$$a_k^t = \sum_{h=1}^{H} w_{fk}\, b_f^t + w_{bk}\, b_b^t \quad (۱۶۱)$$

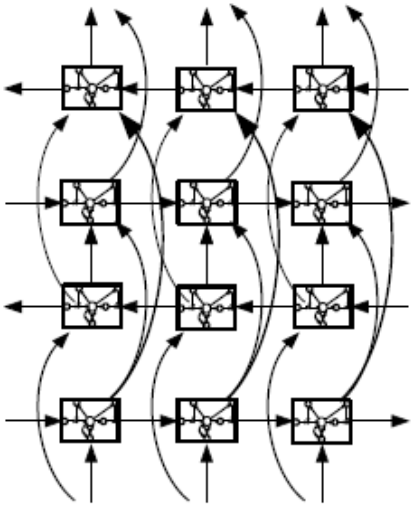
$$b_k^t = f_{softmax}(a_k^t) \quad (۱۶۲)$$

۴-۴-۴- شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه

در این بخش به توضیح پیرامون ساختار، آموزش و نحوه ارزیابی شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه [۵, ۶] می‌پردازیم. همان‌طور که پیش‌تر نیز گفته شد شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه شامل چندین لایه پنهان می‌باشد که هر لایه پنهان شامل دو لایه بازگشتی پیش‌رو و رو به عقب با بلوک‌های حافظه LSTM می‌باشد. در این شبکه دنباله ورودی به‌ترتیب از گام زمانی اول تا آخرین گام زمانی به لایه‌های بازگشتی پیش‌رو و از گام زمانی آخر تا گام زمانی اول به لایه‌های بازگشتی رو به عقب داده می‌شود که این امر سبب می‌گردد که بر خلاف شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه که خروجی شبکه در هر گام زمانی تنها به ورودی فعلی و ورودی‌های قبلی وابسته است، خروجی شبکه در هر گام زمانی به کل دنباله ورودی وابسته باشد و کارایی شبکه افزایش یابد.

۴-۴-۴-۱- ساختار

شبکه عصبی DBLSTM از روی هم قرار دادن تعدادی شبکه BLSTM حاصل می‌گردد. به‌عبارت دیگر هر لایه پنهان این شبکه، شامل لایه پیش‌رو و رو به عقب شبکه BLSTM می‌باشد. در این شبکه هر لایه پیش‌رو یا رو به عقب، از خروجی بلوک‌های حافظه لایه پیش‌رو و رو به عقب سطح زیرین خود ورودی دریافت می‌کند. شکل ۴-۸ ساختار شبکه عصبی عمیق دوطرفه حافظه کوتاه مدت ماندگار با دو لایه پنهان را نشان می‌دهد.



شکل ۴-۸) ساختار شبکه عصبی عمیق دوطرفه حافظه کوتاه مدت ماندگار [۵]

۴-۴-۴-۲- آموزش شبکه

آموزش شبکه شامل دو مرحله پیش‌رو و پس‌رو می‌باشد. در ادامه هر یک از این گام‌ها توضیح داده خواهد شد.

فرض کنید مجموعه آموزش شامل تعدادی دنباله باشد بطوریکه هر دنباله‌ی ورودی X^T و دنباله هدف متناظر آن یعنی Z^T، به فرم زیر باشد:

$$X^T = \{x_1. x_2. \dots . x_{T-1}. x_T\} \quad (۱۶۳)$$

$$Z^T = \{z_1. z_2. \dots . z_{T-1}. z_T\} \quad (۱۶۴)$$

مرحله پیش‌رو

در مرحله پیش‌رو آموزش شبکه، ابتدا فعال‌سازهای مربوط به نرون‌های پیش‌رو و رو به عقب لایه اول محاسبه می‌گردد. سپس فعال‌سازهای این دو لایه به عنوان ورودی نرون‌های پیش‌رو و رو به عقب لایه دوم در نظر گرفته می‌شود. به‌عبارت دیگر همان‌طور که در شکل ۴-۶ دیده می‌شود نرون‌های هر لایه پیش‌رو و رو به عقب فعال‌سازهای نرون‌های پیش‌رو و رو به عقب لایه زیرین را به عنوان ورودی دریافت می‌کند و این روند برای لایه‌های بالاتر تکرار می‌شود.

اگر تعداد لایه‌های پنهان برابر L باشد، هر لایه پنهان را شامل دو لایه پیش‌رو و رو به عقب در نظر بگیرید و برای اجرای مرحله پیش‌رو، گام‌های زیر را به‌ترتیب مشخص شده اجرا نمایید.

۱. برای لایه پنهان $l = 1$ تا $l = L$ گام‌های (۱-۱) و (۱-۲) را به‌ترتیب اجرا کنید.

۱-۱. مرحله پیش‌رو لایه پیش‌رو که در بخش ۲-۳-۴-۴ شرح داده شده است را اجرا نمایید.

۱-۲. مرحله پیش‌رو لایه رو به عقب که در بخش ۲-۳-۴-۴ شرح داده شده است را اجرا نمایید.

۲. محاسبه فعال‌سازهای لایه خروجی

برای گام‌های زمانی $t = 1$ تا $t = T$ فعال‌ساز نرون‌های لایه خروجی را به‌کمک رابطه‌های (۱۶۵) و (۱۶۶) محاسبه نمایید. در این روابط b_b^t و b_f^t به‌ترتیب به فعال‌ساز نرون‌های لایه رو به عقب و لایه پیش‌رو آخرین لایه پنهان در گام زمانی t اشاره می‌کنند. همچنین w_{bk} و w_{fk} به‌ترتیب به وزن بین آخرین لایه رو به عقب و خروجی و وزن بین آخرین لایه پیش‌رو و خروجی اشاره می‌کند.

$$a_k^t = \sum_{h=1}^{H} w_{fk}\, b_f^t + w_{bk}\, b_b^t \quad (۱۶۵)$$

$$b_k^t = f_{softmax}(a_k^t) \quad (۱۶۶)$$

مرحله پس‌رو

در مرحله پس‌رو از آموزش شبکه، خطای نرون‌های لایه خروجی پس انتشار می‌یابد. برای اجرای مرحله پس‌رو، مراحل زیر را به‌ترتیب مشخص شده اجرا نمایید.

۱. برای هر لایه پنهان $l = L$ تا $l = 1$ گام‌های زیر را به‌ترتیب اجرا کنید.

۱-۱. مرحله پس‌رو لایه پیش‌رو که در بخش ۲-۳-۴-۴ شرح داده شده است را اجرا نمایید.

۱-۲. مرحله پس‌رو لایه رو به عقب که در بخش ۲-۳-۴-۴ شرح داده شده است را اجرا نمایید.

۲. تمامی وزن‌ها را با استفاده از رابطه زیر به‌روز رسانی کنید.

$$w_{ij} = w_{ij} + \sum_{t=1}^{T} \Delta w_{ij}^{t} \qquad (۱۶۷)$$

۴-۴-۴-۳- ارزیابی شبکه

فرض کنید مجموعه تست شامل تعدادی دنباله باشد بطوریکه هر دنباله‌ی ورودی X^T و دنباله هدف متناظر آن یعنی Z^T، به فرم زیر باشد:

$$X^T = \{x_1 . x_2 . \dots . x_{T-1} . x_T\} \qquad (۱۶۸)$$

$$Z^T = \{z_1 . z_2 . \dots . z_{T-1} . z_T\} \qquad (۱۶۹)$$

به‌منظور ارزیابی شبکه، گام‌های زیر را به‌ترتیب مشخص شده اجرا نمایید.

۱. برای لایه پنهان $l = 1$ تا $l = L$ گام‌های زیر را به‌ترتیب اجرا کنید.

۱-۱. مرحله پیش‌رو لایه پیش‌رو که در بخش ۲-۳-۴-۴ شرح داده شده است را اجرا نمایید.

۱-۲. مرحله پیش‌رو لایه رو به عقب که در بخش ۲-۳-۴-۴ شرح داده شده است را اجرا نمایید.

۲. محاسبه فعال‌سازهای لایه خروجی

برای گام‌های زمانی $t = 1$ تا $t = T$ فعال‌ساز نرون‌های لایه خروجی را به‌کمک رابطه‌های (۱۷۰) و (۱۷۱) محاسبه نمایید. در این روابط b_b^t و b_f^t به‌ترتیب به فعال‌ساز نرون‌های لایه رو به عقب و لایه پیش‌رو آخرین لایه پنهان در گام زمانی t اشاره می‌کنند. همچنین w_{bk} و w_{fk} به‌ترتیب به وزن بین آخرین لایه رو به عقب و خروجی و وزن بین آخرین لایه پیش‌رو و خروجی اشاره می‌کنند.

$$a_k^t = \sum_{h=1}^{H} w_{fk}\, b_f^t \; + \; w_{bk}\, b_b^t \qquad (۱۷۰)$$

$$b_k^t = f_{softmax}(a_k^t) \quad (۱۷۱)$$

۴-۵- برچسب گذاری دنباله

هدف از برچسب گذاری دنباله، اختصاص دادن دنباله‌ای از برچسب‌ها به دنباله‌ای از ورودی‌ها می‌باشد. در حالت کلی، برچسب گذاری دنباله می‌تواند به سه دسته تقسیم گردد که در ادامه به توضیح هر دسته می‌پردازیم.

۱. طبقه‌بندی دنباله[۱]: در این نوع برچسب گذاری که در واقع ساده‌ترین نوع برچسب گذاری دنباله نیز می‌باشد، کل دنباله ورودی تنها به یک کلاس اختصاص داده می‌شود و به‌عبارت دیگر طول دنباله خروجی برابر یک خواهد بود. یک مثال از این نوع برچسب گذاری تشخیص نام بیماری‌ها می‌باشد که هر فایل صوتی مطابق با نام یک بیماری خاص است و کل دنباله ورودی متناظر با یک برچسب خواهد بود.
۲. طبقه‌بندی بخش[۲]:در این نوع برچسب گذاری هر قسمت از دنباله ورودی متناظر با یک برچسب خواهد بود و به‌عبارتی دیگر دنباله خروجی شامل چندین برچسب خواهد بود. یک مثال از این نوع برچسب گذاری، برچسب گذاری فریم‌های سیگنال صوتی می‌باشد.

۳. طبقه‌بندی زمانی[۳]: این نوع برچسب گذاری پیچیده ترین و کلی‌ترین حالت برچسب گذاری است و تنها محدودیت لحاظ شده در آن این است که طول دنباله برچسب حداکثر می‌تواند به اندازه طول دنباله ورودی باشد. بنابراین در این نوع برچسب گذاری، تهی بودن دنباله خروجی نیز مجاز خواهد بود. یک مثال از این نوع برچسب گذاری، تشخیص دنباله واج خروجی متناظر با یک سیگنال صوتی می‌باشد.

۴-۶- طبقه‌بند زمانی پیوندگرا

یکی از محدودیت‌های استفاده از شبکه‌های عصبی برای بازشناسی گفتار برچسب گذاری مجزای هر فریم از سیگنال ورودی توسط شبکه عصبی می‌باشد. بدین معنی که شبکه به‌جای تولید دنباله واج متناظر با سیگنال ورودی، برچسب هر فریم را بصورت جداگانه تشخص می‌دهد. بنابراین برای دستیابی به دنباله واج متناظر با سیگنال صوتی، نیاز به استفاده از الگویتم‌های پس پردازش[۴] جهت استخراج دنباله واج متناظر با سیگنال ورودی از روی خروجی‌های مربوط به هر فریم که

[۱] Sequence Classification
[۲] Segment Classification
[۳] Temporal Classification
[۴] Post Processing

توسط شبکه عصبی به‌دست آمده است می‌باشد. یکی از راه حل‌ها برای این مساله جایگزین کردن لایه خروجی متداول شبکه عصبی با لایه CTC می‌باشد [۴۳]. در الگوریتم CTC تنها نحوه محاسبه دلتای نرون‌های لایه خروجی تغییر می‌کند و ساختار شبکه می‌تواند به صورت یک‌طرفه، دوطرفه یا عمیق باشد

لایه خروجی CTC در شبکه‌های عصبی بازگشتی، برای کارهای طبقه‌بندی زمانی که در بخش قبلی توضیح داده شد، مورد استفاده قرار می‌گیرد. به‌عبارت دیگر، کاربرد این الگوریتم برچسب گذاری دنباله‌هایی است که در آن‌ها نگاشت بین دنباله ورودی و دنباله هدف مشخص نمی‌باشد. بنابراین در این الگوریتم احتیاجی به پس پردازش خارجی به‌منظور استخراج دنباله برچسب از روی خروجی شبکه عصبی نمی‌باشد و شبکه به‌جای تولید برچسب متناظر با هر فریم دنباله واج متناظر با کل سیگنال را تولید می‌کند. الگوریتم طبقه‌بند زمانی پیوندگرا شامل دو الگوریتم پیش‌رو–پس‌رو و رمز گشایی می‌باشد. الگوریتم پیش‌رو–پس‌رو در بخش آموزش و الگوریتم رمز گشایی در بخش تولید دنباله واج متناظر با سیگنال ورودی از روی خروجی شبکه مورد استفاده قرار می‌گیرد که در ادامه به توضیح آن‌ها می‌پردازیم.

۴-۶-۱-۱- الگوریتم آموزش

اگر A مجموعه برچسب‌های مجاز در دنباله خروجی باشد، تعداد نرون‌های لایه CTC یک واحد بیشتر از تعداد اعضای مجموعه A خواهد بود. هر نرون لایه CTC معادل یکی از برچسب‌های موجود در مجموعه A خواهد بود و نرون آخر معادل برچسب تهی می‌باشد. بنابر این با توجه به این‌که فعال‌ساز مورد استفاده در لایه CTC تابع $softmax$ می‌باشد، مقدار فعال‌ساز $|A|$ نرون اول این لایه، معادل احتمال تولید برچسب مربوط به آن نرون توسط شبکه در گام زمانی متناظر است و مقدار فعال‌سازی آخرین نرون بیانگر احتمال تولید خروجی خالی[۱] توسط شبکه می‌باشد. بنابراین مجموعه مجاز برچسب‌های خروجی به‌صورت $A' = \{A \cup Blank\}$ خواهد بود. اگر z دنباله هدف با طول U باشد، الگوریتم CTC از دنباله z' به طول $U' = 2U + 1$ که با اضافه نمودن برچسب خالی به ابتدا، انتها و بین هر دو برچسب متوالی حاصل می‌گردد، استفاده می‌کند. تعریف برچسب تهی این امکان را به شبکه می‌دهد که شبکه بتواند دو برچسب یکسان را به صورت متوالی تولید نماید. علاوه بر این، در برخی کاربردها مانند بازشناسی گفتار در اغلب موارد بین کلمات وقفه وجود دارد و هیچ واجی ادا نمی‌شود.

اگر A'^T مجموعه تمام دنباله‌های به طول T روی مجموعه A' باشد، به هر عضو مجمو عه A'^T یک مسیر می‌گوییم و آن‌را با نماد π نمایش می‌دهیم. فرض کنید که y_k^t احتمال این است که شبکه در گام زمانی t برچسب k را تولید نماید. بنابراین با فرض مستقل بودن خروجی شبکه در گام‌های زمانی متفاوت احتمال این‌که شبکه مسیر π را تولید کند برابر رابطه (۱۷۲) خواهد بود:

$$p(\pi|x) = \prod_{t=1}^{t=T} y_{\pi_t}^t \qquad (۱۷۲)$$

[۱] Blank

در مرحله بعد نگاشت $\mathrm{F}: A'^{T} \longrightarrow A^{\leq T}$ را از مجموعه مسیرهای به طول T روی مجموعه برچسب‌های A' به مسیرهایی که طول آن‌ها حداکثر برابر طول T روی مجموعه برچسب‌های A می‌باشد با قوانین زیر تعریف می‌کنیم:

۱. ابتدا تمام برچسب‌های تکراری را حذف می‌کنیم.

۲. سپس تمام برچسب‌های خالی را حذف می‌کنیم.

اگر مجموعه $V(t.u)$ را به‌صورت $V(t.u) = \left\{\pi \in A'^{T} : \mathrm{F}(\pi) = z_{1:u/2} \,.\, \pi_t = z'_u\right\}$ تعریف کنیم (یعنی مجموعه تمامی مسیرهای به طول t که پس از اعمال نگاشت F روی آن‌ها زیر دنباله به طول $u/2$ از پیشوند z حاصل می‌گردد)، آنگاه متغیر پیش‌رو $\alpha(t.u)$ برابر جمع احتمالاتی تمامی مسیرهای عضو مجموعه $V(t.u)$ خواهد بود.

همچنین اگر مجموعه $W(t.u)$ را به‌صورت $W(t.u) = \left\{\pi \in A'^{T-t} : \mathrm{F}(\hat{\pi} + \pi) = z \,.\, \forall\, \hat{\pi} \in V(t.u)\right\}$ تعریف کنیم، آنگاه متغیر پس‌رو $\beta(t.u)$ برابر جمع احتمالاتی تمامی مسیرهای عضو مجموعه $W(t.u)$ خواهد بود.

با در نظر گرفتن توضیحات ارائه شده، در ادامه مراحل آموزش الگوریتم را شرح می‌دهیم. نمادهای به‌کار رفته در الگوریتم CTC در جدول ۴-۲ آورده شده است.

جدول ۴-۲) نمادهای به‌کار رفته در الگوریتم CTC

تعریف	نماد
خروجی شبکه متناظر با k امین برچسب مجموعه A' در گام زمانی t	y_k^t
دنباله برچسب هدف	z
دنباله برچسب هدف که به آن برچسب $Blank$ اضافه شده	z'
مجموعه برچسب‌های مجاز به‌همراه $Blank$	A'
مجموعه تمام دنباله‌های به طول T روی مجموعه A'	A'^{T}
مسیر به طول T (هر عضو مجموعه) A'^{T}	$\pi \in A'^{T}$
متغیر U	$\|z\|$
متغیر U'	$\|z'\|$
دلتای نرون k اُم لایه CTC در گام زمانی t	δ_k^t

برای هر سیگنال آموزش مراحل زیر را به‌ترتیب تکرار کنید:

۱. در ابتدا، انتها و بین هر دو برچسب دنباله هدف برچسب $Blank$ قرار دهید و آن را z' نام‌گذاری کنید.

۲. متغیر α را مقداردهی اولیه نمایید:

$$\alpha(1.1) = y_b^1 \quad (۱۷۳)$$

$$\alpha(1.2) = y_{z_1}^1 \quad (۱۷۴)$$

$$\alpha(1.u) = 0 \; \forall \, u > 2 \quad (۱۷۵)$$

۳. با استفاده از رابطه‌ی (۱۷۶) برای تمام گام‌های زمانی $t = 2$ تا $t = T$ متغیر α را به ازای $u = 1$ تا $u = U'$ محاسبه نمایید:

$$\alpha(t.u) = \begin{cases} 0 & if \; u < U' - 2(T-t) - 1 \\ y_{z'_u}^t \sum_{i=f(u)}^{u} \alpha(t-1.i) & otherwise \end{cases} \quad (۱۷۶)$$

که در رابطه بالا، $f(u)$ برابر است با:

$$f(u) = \begin{cases} u-1 & if \; z'_u = blank \; or \; z'_{u-2} = z'_u \\ u-2 & otherwize \end{cases} \quad (۱۷۷)$$

۴. متغیر β را مقداردهی اولیه نمایید:

$$\beta(T.U') = 1 \quad (۱۷۸)$$

$$\beta(T.U' - 1) = 1 \quad (۱۷۹)$$

$$\beta(T.u) = 0 \; \forall \, u < U' - 1 \quad (۱۸۰)$$

۵. با استفاده از رابطه‌ی (۱۸۱) برای تمام گام‌های زمانی $t = T$ تا $t = 1$ متغیر β را به‌ازای $u = U'$ تا $u = 1$ محاسبه نمایید:

$$\beta(t.u) = \begin{cases} 0 & if \; u > 2t \\ \sum_{i=u}^{g(u)} \beta(t+1.i) \, y_{z'_i}^{t+1} & otherwise \end{cases} \quad (۱۸۱)$$

که در رابطه بالا، $g(u)$ برابر است با:

$$g(u) = \begin{cases} u+1 & if \; z'_u = blank \; or \; z'_{u+2} = z'_u \\ u+2 & otherwize \end{cases} \quad (۱۸۲)$$

۶. احتمال مشاهده دنباله برچسب صحیح به‌شرط سیگنال ورودی را به کمک رابطه (۱۸۳) محاسبه کنید:

$$P(z|x) = \sum_{u=1}^{u=|z'|} \alpha(t.u) \, \beta(t.u) \quad (۱۸۳)$$

۷. مجموعه $B(z.k)$ را معادل مجموعه موقعیت‌هایی که برچسب k در z' قرار گرفته است قرار دهید. بنابراین $B(z.k)$ به‌صورت رابطه (۱۸۴) تعریف می‌گردد:

$$B(z.k) = \{u: z'_u = k\} \quad (۱۸۴)$$

۸. خطای نرون‌های لایه خروجی CTC را محاسبه نمایید:

$$\delta_k^t = y_k^t - \frac{1}{P(z|x)} \sum_{u \in B(z.k)} \alpha(t.u)\, \beta(t.u) \quad (۱۸۵)$$

۴-۶-۱-۲- الگوریتم رمز گشایی

پس از آموزش شبکه با الگوریتم CTC باید دنباله واج متناظر با سیگنال ورودی از خروجی شبکه استخراج شود. برای این منظور از الگوریتم بهترین مسیر[۱] استفاده می‌نماییم. به‌منظور رمز گشایی گام‌های زیر را دنبال نمایید.

۱. در هر گام زمانی نرون با بیشترین مقدار احتمال را انتخاب نمایید.

۲. برچسب‌های تکراری پشت سر هم را حذف نمایید.

۳. برچسب Blank را حذف نمایید.

۴-۷- شبکه عصبی باور عمیق

شبکه عصبی باور عمیق [۱۷]، یک شبکه عصبی با ساختار عمیق می‌باشد که هر لایه آن یک ماشین بولتزمن محدود [۹۸] است. هر ماشین بولتزمن محدود یک مدل گرافیکی بدون جهت است که اتصالات بین واحدهای مخفی و همچنین اتصالات بین واحدهای مشاهده پذیر قطع شده است و یک توزیع احتمالاتی را روی مجموعه داده‌های ورودی مدل می‌کند. یکی از کاربردهای این شبکه در استخراج ویژگی می‌باشد که میتواند از داده‌های برچسب نخورده سطح بالایی از ویژگی‌ها را استخراج نماید [۹۹]. در ادامه ابتدا ساختار و نحوه آموزش RBM را توضیح می‌دهیم و پس از نحوه استفاده از آن‌را در شبکه باور عمیق جهت استخراج ویژگی شرح می‌دهیم.

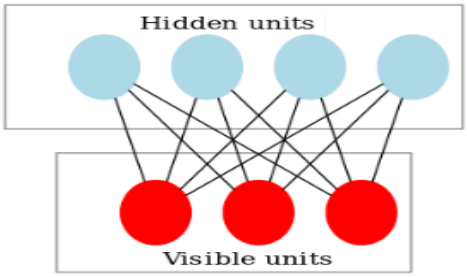
۴-۷-۱- ماشین بولتزمن محدود

هر ماشین بولتزمن محدود یک مدل گرافیکی بدون جهت است که اتصالات بین واحدهای مخفی و همچنین اتصالات بین واحدهای مشاهده پذیر قطع شده است و یک توزیع احتمالاتی را روی مجموعه داده‌های ورودی مدل می‌کند. در این بخش ابتدا ساختار و سپس الگوریتم آموزش RBM را توضیح می‌دهیم.

[۱] Best Path Decoding

۴-۷-۱-۱- ساختار

همان‌طور که گفته شد ماشین بولتزمن محدود نوعی ماشین بولتزمن است که در آن اتصالات بین واحدهای مخفی و همچنین واحدهای مشاهده‌پذیر قطع شده است. ماشین بولتزمن نیز نوعی مدل گرافیکی بدون جهت با وزن‌های متقارن می‌باشد. هر ماشین بولتزمن شامل دولایه مشاهده‌پذیر و پنهان می‌باشد. شکل ۴-۹ ساختار ماشین بولتزمن محدود را نمایش می‌دهد.



شکل ۴-۹) ساختار ماشین بولتزمن محدود

۴-۷-۱-۲- آموزش

ماشین بولتزمن محدود استاندارد، شامل واحدهای مشاهده‌پذیر و پنهان با مقادیر باینری می‌باشد. اگر وزن بین واحد مشاهده‌پذیر i و واحد پنهان j برابر w_{ij} باشد و بایاس برای واحد مشاهده‌پذیر i و واحد پنهان j به‌ترتیب برابر a_i و b_j باشد، انرژی حالت $(v.h)$برابر رابطه (۱۸۶) خواهد بود.

$$E(v.h) = -\sum_i a_i\, v_i - \sum_i b_j\, h_j - \sum_i \sum_j v_i w_{ij}\, h_j \qquad (۱۸۶)$$

RBM به هر حالت ممکن مقادیر بردارهای مشاهده‌پذیر و پنهان یک مقدار احتمال نسبت می‌دهد که مقدار آن از رابطه (۱۸۷) به‌دست می‌آید.

$$P(v.h) = \frac{1}{Z}\, e^{-E(v.h)} \qquad (۱۸۷)$$

که در این رابطه Z یک مقدار ثابت است و برابر است با مجموع مقادیر $e^{-E(v.h)}$. در واقع مقدار Z نقش نرمال‌کننده را دارد تا مجموع مقادیر احتمال برابر یک شود. احتمالی که مدل به بردار مشاهده‌پذیر v نسبت می‌دهد برابر رابطه (۱۸۸) خواهد بود.

$$P(v) = \frac{1}{Z} \sum_{h} e^{-E(v.h)} \quad (۱۸۸)$$

با توجه به این‌که هیچ اتصالی بین واحدهای واحدهای مشاهده‌پذیر وجود ندارد، می‌توان فرض کرد که این واحدها به‌شرط واحدهای پنهان داده شده از یکدیگر مستقل هستند و به‌طور مشابه با توجه به این‌که واحدهای پنهان به یکدیگر متصل نیستند بنابراین این واحدها به‌شرط واحدهای مشاهده‌پذیر از یکدیگر مستقل هستند. در نتیجه اگر m واحد مشاهده‌پذیر و n واحد پنهان داشته باشیم، احتمال بردار مشاهده‌پذیر v به‌شرط بردار پنهان hو بالعکس را می‌توان به کمک رابطه‌های (۱۸۹) و (۱۹۰) محاسبه کرد.

$$P(v|h) = \prod_{i=1}^{m} P(v_i|h) \quad (۱۸۹)$$

$$P(h|v) = \prod_{j=1}^{n} P(h_j|v) \quad (۱۹۰)$$

احتمال فعال‌شدن (یک شدن مقدار فعال‌سازی) هریک از واحدهای مشاهده‌پذیر به‌شرط داشتن مقادیر واحدهای پنهان و همچنین احتمال فعال‌شدن هر یک از واحدهای پنهان به‌شرط داشتن مقادیر واحدهای مشاهده‌پذیر به کمک رابطه‌های (۱۹۲) و (۱۹۱) محاسبه می‌گردد. در این روابط σ نماد تابع سیگموید[۱] می‌باشد.

$$P(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^{m} w_{ij}\, v_i) \quad (۱۹۱)$$

$$P(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^{n} w_{ij}\, h_j) \quad (۱۹۲)$$

الگوریتمی که به‌صورت معمول جهت آموزش RBM مورد استفاده قرار می‌گیرد، واگرایی متقابل[۲] (CD) نام دارد [۱۰۰] که در ادامه مراحل آن را شرح می‌دهیم.

برای بردار آموزشی vمراحل زیر را به‌ترتیب مشخص شده دنبال کنید:

۱. با استفاده از رابطه (۱۹۱) احتمال فعال شدن واحدهای پنهان را محاسبه نمایید. مقدار فعال‌سازی هر یک از نرون‌های لایه پنهان را در صورتی‌که از حد آستانه[۳] بیشتر بود برابر یک و در غیر این‌صورت برابر صفر قرار دهید.

[۱] Sigmoid

[۲] Contrastive Divergence (CD)

[۳] Threshold

۲. ضرب خارجی[۱] v و h را به کمک رابطه (۱۹۳) محاسبه نمایید و آن‌را گرادیان مثبت[۲] بنامید.

$$pg = v\,h^T \quad (۱۹۳)$$

۳. با استفاده از رابطه (۱۹۲) احتمال فعال شدن واحدهای مشاهده‌پذیر را محاسبه نمایید و بردار به‌دست آمده را v' بنامید. برای این منظور مقدار فعال‌سازی هر یک از نرون‌های لایه مشاهده‌پذیر را در صورتی‌که از مقدار حد آستانه بیشتر بود برابر یک و در غیر این‌صورت برابر صفر قرار دهید.

۴. دوباره با استفاده از رابطه (۱۹۱) احتمال فعال شدن واحدهای پنهان را با در نظر گرفتن بردار v' به عنوان مقادیر لایه مشاهده‌پذیر محاسبه نمایید و آن را h' نام‌گذاری کنید.

۵. ضرب خارجی v' و h' را به کمک رابطه (۱۹۴) محاسبه نمایید و آن را گرادیان منفی[۳] بنامید.

$$ng = v'\,h'^T \quad (۱۹۴)$$

۶. اگر نرخ یادگیری برابر α باشد تغییرات وزن را به کمک رابطه‌های زیر محاسبه کنید.

$$\Delta W = \alpha(v\,h^T - v'\,h'^T) \quad (۱۹۵)$$

$$\Delta b = \alpha(h - h') \quad (۱۹۶)$$

$$\Delta a = \alpha(v - v') \quad (۱۹۷)$$

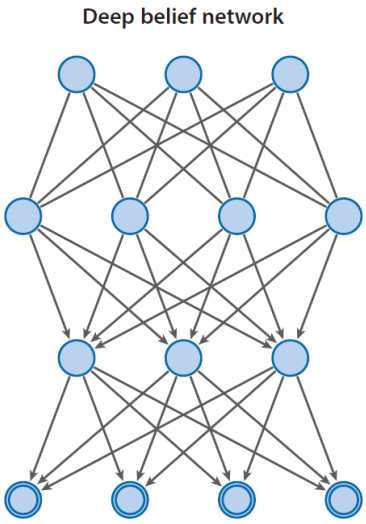
۴-۷-۲- ساختار شبکه باور عمیق

شبکه عصبی باور عمیق یک شبکه عصبی چند لایه می‌باشد که هر لایه آن یک ماشین بولتزمن محدود است [۱۰۱]. به‌عبارت دیگر با روی هم قرار دادن تعدادی RBM یک شبکه باور عمیق حاصل می‌گردد. شکل ۴-۱۰ ساختار شبکه DBN را نمایش می‌دهد.

[۱] Outer Prouduct

[۲] Positive Gradient

[۳] Negative Gradient

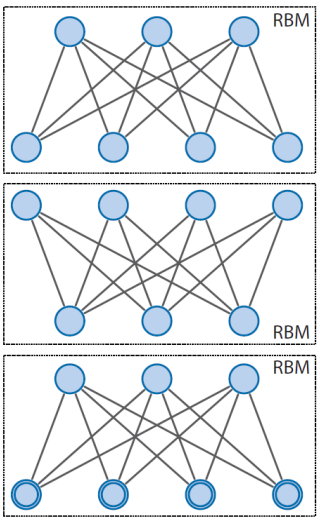


شکل ۴-۱۰) ساختار شبکه DBN [۱۰۱]

۴-۷-۳- آموزش شبکه باور عمیق

الگوریتم آموزش این شبکه یک الگوریتم حریصانه[۱] است و لایه‌های RBM به‌ترتیب از لایه اول تا آخرین لایه آموزش می‌بینند. بدین معنی که ابتدا ماشین بولتزمن محدود زیرین با پارامترهای W^1 آموزش داده می‌شود. سپس وزن‌های لایه دوم با مقدار $W^2 = W^{1^T}$ مقدار دهی اولیه می‌گردد تا این اطمینان حاصل شود که شبکه دو لایه حداقل به‌میزان شبکه یک لایه کارایی دارد. سپس مقادیر خروجی لایه پنهان اول به عنوان داده ورودی برای لایه پنهان دوم در نظر گرفته شده و لایه دوم نیز آموزش می‌بینند. در صورت وجود بیشتر از دو لایه این روند تکرار می‌گردد. البته در حالت کلی احتیاجی نیست که اندازه ماتریس هر لایه با لایه زیرین یکسان باشد. شکل ۴-۱۱ یادگیری حریصانه یک DBN سه لایه را نمایش می‌دهد.

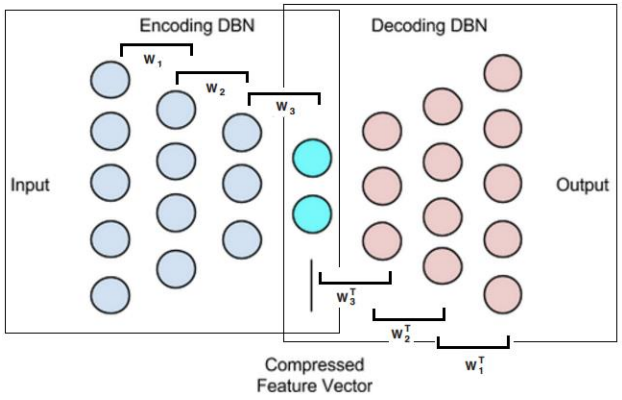
[۱] Greedy



شکل ۴-۱۱) یادگیری حریصانه DBN سه لایه [۱۰۱]

۴-۷-۴- استخراج ویژگی با شبکه باور عمیق

جهت استخراج ویژگی با استفاده از شبکه DBN، از شبکه باور عمیق خود رمزگذار استفاده می‌کنیم [۹۷]. شبکه باور عمیق خود رمزگذار شامل دو بخش رمزگذار و رمزگشا می‌باشد که بخش رمزگذار آن جهت کاهش ابعاد داده ورودی و به‌عبارت دیگر کد کردن آن است و بخش رمزگشا جهت ساخت دوباره داده ورودی از روی داده‌ی کد شده می‌باشد. شکل ۴-۱۱ ساختار یک شبکه شبکه باور عمیق خود رمزگذار را نمایش می‌دهد. همان‌طور که در شکل دیده می‌شود، در شبکه تعداد نرون‌های لایه ورودی برابر تعداد نرون‌های لایه خروجی می‌باشد.



شکل ۴-۱۲) ساختار شبکه DBN Auto-Encoder

به‌منظور آموزش شبکه، وزن‌های اولیه هر دو بخش رمزگشا و رمزگذار را یکسان و برابر وزن‌های به‌دست آمده از آموزش شبکه DBN که در قسمت قبل توضیح داده شد قرار می‌دهیم. سپس داده‌ها در جهت رو به جلو به شبکه داده می‌شود و خطای محاسبه شده در لایه خروجی پس انتشار [۹۰] می‌یابد. پس از آموزش شبکه به‌منظور استخراج ویژگی، داده‌ها به بخش رمزگذار شبکه داده می‌شود و خروجی این بخش همان ویژگی‌های مورد نیاز می‌باشد.

فصل پنجم: نتایج و ارزیابی‌ها

۵-۱- مقدمه

در این فصل در ابتدا مجموعه دادگان مورد استفاده و نحوه استخراج ویژگی از سیگنال‌ها توضیح داده می‌شود و پس از آن نتایج حاصل از پیاده‌سازی شبکه‌های عصبی حافظه کوتاه مدت ماندگار، حافظه کوتاه مدت ماندگار دوطرفه، حافظه کوتاه مدت ماندگار عمیق یک‌طرفه و همچنین حافظه کوتاه مدت ماندگار عمیق دوطرفه ارائه خواهد گردید.

۵-۲- مجموعه دادگان

مجموعه داده‌های مورد استفاده مجموعه فارس‌دات [۱۵] می‌باشد. این مجموعه در سال ۱۳۷۵ توسط پژوهشکده پردازش هوشمند علائم تهیه گردید. فارس‌دات شامل ۳۸۶ جمله متفاوت می‌باشد که توسط ۳۰۰ گوینده با ۱۰ لهجه متفاوت بیان شده است. هر گوینده حدود ۲۰ جمله را در محیط آکوستیکی بیان کرده است و این جملات با نرخ فرکانس ۲۲۰۵۰ هرتز ضبط گردیده است. این مجموعه شامل ۶۰۸۰ سیگنال صوتی می‌باشد. نحوه استفاده از این داده‌ها جهت پیاده‌سازی مدل به‌صورت زیر می‌باشد:

۱. دادگان آموزش[۱]: حدود ۸۰ ٪ داده‌های این مجموعه یعنی تعداد ۴۸۶۴ سیگنال جهت آموزش شبکه مورد استفاده قرار گرفته است.

[۱] Train Data

۲. دادگان تست[۲]: ۱۸٪ کل داده‌ها یعنی ۱۰۹۵ سیگنال نیز جهت تست شبکه مورد استفاده قرار گرفته است.

۳. دادگان ارزیابی[۳]: حدود ۲ درصد از داده‌های این مجموعه یعنی تعداد ۱۲۱ سیگنال به عنوان مجموعه ارزیابی مورد استفاده قرار گرفته است.

۵-۳- معیار ارزیابی

۵-۳-۱- دقت در سطح فریم

برای ارزیابی دقت شبکه در سطح فریم، تعیین می‌کنیم که شبکه برچسب چه تعدادی از فریم‌ها را به درستی تشخیص داده است. برای این منظور از رابطه (۱) استفاده می‌نماییم.

$$\text{دقت در سطح فریم} = \frac{\text{تعداد فریم‌های درست تشخیص داده شده}}{\text{تعداد کل فریم‌ها}} * 100 \qquad (۱)$$

۵-۳-۲- دقت در سطح واج

انواع خطاهایی که شبکه هنگام تولید دنباله واج متناظر با سیگنال ورودی تولید می‌کند عبارت است از:

۱. خطای حذف[۴]: این خطا هنگامی پدیدار می‌گردد که شبکه در دنباله خروجی کاراکتری را تولید کند که در دنباله هدف وجود ندارد. به‌عبارت دیگر این کاراکتر باید حذف گردد.

۲. خطای درج[۵]: این خطا هنگامی ایجاد می‌شود که کاراکتری در دنباله هدف وجود دارد که شبکه آن را در دنباله خروجی تولید نکرده است

۳. خطای جابجایی[۶]: این خطا هنگامی ایجاد می‌شود که برای تطابق پیدا کردن دنباله خروجی شبکه با دنباله هدف باید کاراکتر تشخیص داده شده توسط شبکه با کاراکتری دیگر تعویض شود.

اگر طول دنباله تشخیص داده شده توسط شبکه برابر N باشد، در این‌صورت نرخ خطا در سطح واج با استفاده از رابطه (۲) به‌دست می‌آید.

$$PER = \frac{Insert + Delete + Substitution}{N} \qquad (۲)$$

---

[۲] Test Set

[۳] Validation Set

[۴] Delete Error

[۵] Insert Error

[۶] Substitution Error

بنابراین دقت در سطح واج باز رابطه (۳) قابل محاسبه می‌باشد.

$$Accuracy = (1 - PER) * 100 \quad (۳)$$

۵-۴- استخراج ویژگی

در این بخش به روش‌های مورد استفاده در این پایان‌نامه جهت استخراج ویژگی از سیگنال‌های مجموعه فارس‌دات می‌پردازیم.

۵-۴-۱- استخراج ویژگی با استفاده از ضرایب کپسترال در مقیاس مل

به‌منظور استخراج ویژگی با استفاده از روش MFCC، ابتدا هر سیگنال را به فریم‌هایی به طول ۱۶ میلی‌ثانیه با میزان هم‌پوشانی ۸ میلی‌ثانیه تبدیل شده است. سپس از هر فریم بعد از پنجره‌گذاری تعداد ۳۹ ضریب MFCC استخراج گردیده است. بنابراین هر فریم به بردار ویژگی به طول ۳۹ تبدیل گردیده است. از آنجایی‌که ورودی شبکه عصبی بردار ویژگی مرتبط با فریم‌ها می‌باشد و در روشMFCC از هر فریم ۳۹ ویژگی استخراج گردیده است. لازم به ذکر است که در این پایان‌نامه جهت استخراج ویژگی‌های MFCC از جعبه ابزار HTK[۷] [۱۰۲] استفاده گردیده است. خلاصه پارامترهای مورد استفاده جهت استخراج ویژگی‌های MFCC در جدول ۵-۱ آمده است.

جدول ۵-۱) پارامترهای مورد استفاده برای استخراج ویژگی‌های MFCC

طول فریم	تعداد فیلترهای مل	نوع پنجره گذاری	میزان هم‌پوشانی فریم‌ها	تعداد ویژگی‌های MFCC
۱۶ میلی ثانیه	۲۶	همینگ	۸ میلی ثنیه	۳۹

۵-۴-۲- استخراج ویژگی با استفاده از شبکه باور عمیق

به‌منظور استخراج ویژگی با استفاده از شبکه باور عمیق ویژگی‌های MFCC هر ۵ فریم متوالی (هر فریم به‌همراه چهار فریم بعدی) داده آموزش را به عنوان داده ورودی جهت آموزش شبکه DBN استفاده می‌کنیم. بنابراین تعداد نرون‌های لایه ورودی شبکه DBN برابر ۱۹۵ می‌باشد. شبکه DBN مورد استفاده شامل ۴ لایه RBM می‌باشد که لایه‌های ۱ تا ۴ به‌ترتیب شامل ۱۰۲۴، ۵۱۲، ۲۵۶ و ۳۹ نرون می‌باشد. همچنین به‌منظور قابل مقایسه بودن نتایج به‌دست آمده با نتایج حاصل از MFCC تعداد نرون‌های آخرین لایه شبکه DBN معادل ۳۹ در نظر گرفته شده است. پس از آموزش DBN،

[۷] Hidden Markov Model Toolkit (HTK)

از مدل به‌دست آمده جهت استخراج ویژگی داده‌های تست و ارزیابی استفاده می‌کنیم. لازم به‌ذکر است که از ساختارهای دیگر DBN از جمله هر فریم به‌همراه ۲ فریم قبل و بعد آن و همچنین هر فریم به‌همراه ۴ فریم قبلی جهت استخراج ویژگی استفاده شده است ولی بهترین نتایج مربوط به حالتی است که هر فریم به‌همراه ۴ فریم بعدی به شبکه DBN داده شد. خلاصه پارامترهای مورد استفاده جهت استخراج ویژگی با شبکه باور عمیق در جدول ۵-۲ آمده است.

جدول ۵-۲) پارامترهای مورد استفاده برای استخراج ویژگی‌ها با استفاده از DBN

پارامتر	مقدار
داده ورودی DBN	ویژگی‌های MFCC هر فریم به‌همراه ۴ فریم بعدی
تعداد نرون‌های لایه ورودی	۱۹۵
تعداد نرون‌های RBM اول	۱۰۲۴
تعداد نرون‌های RBM دوم	۵۱۲
تعداد نرون‌های RBM سوم	۲۵۶
تعداد نرون‌های RBM چهارم	۳۹
تعداد ویژگی‌های استخراج شده	۳۹

۵-۵- پارامترهای موثر بر کارایی شبکه‌ها و نحوه تعیین مقدار آن‌ها

به‌منظور آموزش شبکه ابتدا باید ساختار و مقدار بهینه پارامترهای موثر بر فرآیند یادگیری شبکه تعیین گردند. برای تعیین مقدار بهینه هر پارامتر، باید کارایی شبکه به‌ازای مقادیر مختلف آن پارامتر با فرض ثابت بودن سایر پارامترها ارزیابی گردد. پس از تعیین پارامترهای بهینه، شبکه را با پارامترهای مشخص شده آموزش می‌دهیم و مدل به‌دست آمده را روی داده‌های تست ارزیابی می‌کنیم.

پارامترهای موثر بر آموزش شبکه شامل نرخ یادگیری، تعداد بلوک‌های لایه پنهان و همچنین تعداد لایه‌های پنهان می‌باشد. در ادامه هر یک از این پارامترها را شرح می‌دهیم.

۱. تعداد بلوک‌های لایه پنهان: این پارامتر یکی از پارامترهای بسیار تاثیرگذار در آموزش شبکه می‌باشد. اگر تعداد کم بلوک‌های لایه پنهان باشد باعث کاهش کارایی شبکه می‌گردد و تعداد زیاد بلوک‌ها بار محاسباتی شبکه را بالا برده و از طرفی دیگر تاثیر مثبتی بر کارایی شبکه نخواهد داشت.

۲. نرخ یادگیری: یکی دیگر از پارامترهای موثر بر آموزش شبکه تعیین نرخ مناسب یادگیری است. نرخ یادگیری که مقدار آن عددی در بازه صفر تا یک است، بر روی سرعت و روند همگرایی شبکه تاثیر می‌گذارد. اگر مقدار آن بزرگ باشد شبکه همگرا نخواهد شد و در صورتی که بسیار کوچک باشد فرآیند یادگیری بسیار کند خواهد شد.

۳. تعداد لایه‌های پنهان: یکی دیگر از پارامترهای موثر بر فرآیند یادگیری شبکه تعداد لایه‌های پنهان در شبکه‌های عصبی عمیق می‌باشد. افزایش تعداد لایه‌ها علاوه بر این‌که می‌تواند باعث بهبود دقت شبکه گردد ولی به دلیل افزایش تعداد پارامترهای شبکه روند یادگیری شبکه کندتر می‌گردد.

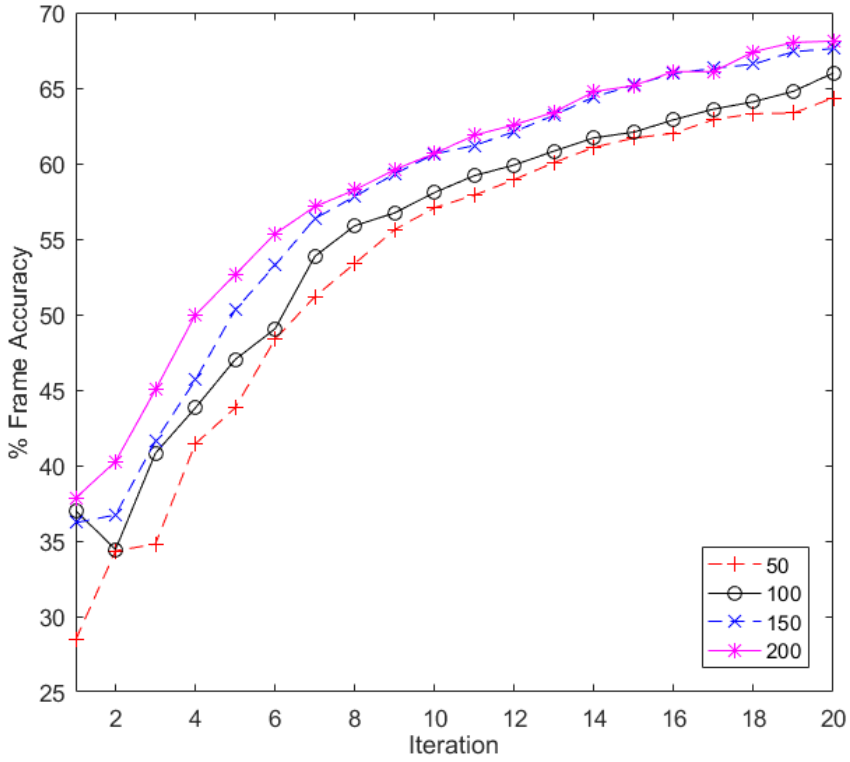
در نتایجی که در ادامه آمده است، ابتدا تاثیر این پارامترها بر روند آموزش شبکه‌ها بررسی شده است و پس از آن نتایج حاصل از آموزش شبکه به‌ازای مقادیر مختلف پارامترها روی مجموعه داده‌های تست آمده است.

۵-۶- نتایج شبکه عصبی حافظه کوتاه مدت ماندگار

۵-۶-۱- تشخیص فریم

۵-۶-۱-۱- تاثیر تعداد بلوک‌های حافظه

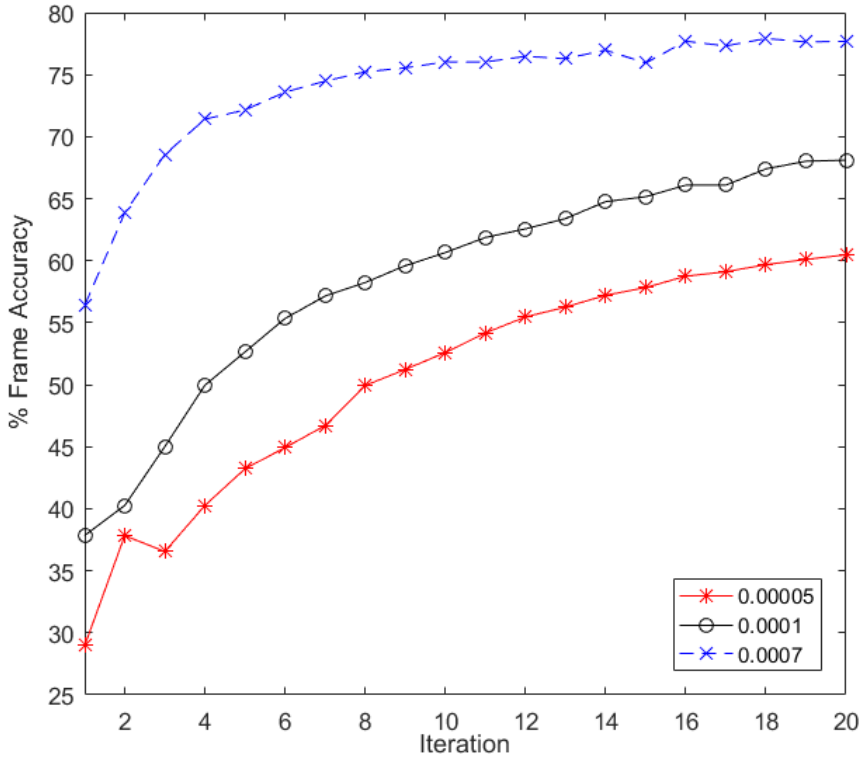
برای بررسی این پارامتر، مقدار نرخ یادگیری برابر مقدار ثابت ۰,۰۰۰۱ در نظر گرفته شده است و شبکه به‌ازای ۵۰، ۱۰۰، ۱۵۰ و ۲۰۰ بلوک حافظه ۲۰ مرحله آموزش داده شده است. همان‌طور که در شکل ۵-۱ دیده می‌شود، بهترین نتیجه مربوط به ۲۰۰ بلوک لایه میانی است.



شکل ۵-۱) دقت تشخیص فریم LSTM به‌ازای نرخ یادگیری ۰٫۰۰۰۱ در ۲۰ مرحله آموزش

۵-۶-۱-۲- تاثیر نرخ یادگیری

به‌منظور بررسی تاثیر نرخ یادگیری، تعداد بلوک‌های لایه پنهان برابر ۲۰۰ قرار داده شده است و شبکه به‌ازای چند نرخ یادگیری مختلف ۲۰ مرحله روی مجموعه فارس‌دات آموزش داده شده است. همان‌طور که در شکل ۵-۲ دیده می‌شود، بهترین نتیجه مربوط به حالتی می‌باشد که نرخ یادگیری برابر ۰٫۰۰۰۷ است.



شکل ۵-۲) دقت تشخیص فریم LSTM به‌ازای ۲۰۰ بلوک حافظه در ۲۰ مرحله آموزش

۵-۶-۱-۳- نتایج شبکه با ویژگی‌های MFCC

جدول ۵-۳ دقت شبکه را روی داده‌های تست به‌ازای پارامترهای مختلف نمایش می‌دهد. همان‌طور که دیده می‌شود، بهترین دقت مربوط به شبکه LSTM با ۲۰۰ بلوک حافظه و نرخ یادگیری ۰٫۰۰۰۷ می‌باشد که این دقت معادل ٪۷۸٫۶ است.

جدول ۵-۳) نتایج دقت LSTM یک‌طرفه در سطح فریم روی داده‌های تست

تعداد بلوک حافظه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
۵۰	۰٫۰۰۰۱	۱۰۱	۷۵٫۱
۱۰۰	۰٫۰۰۰۱	۹۹	۷۷٫۱
۱۵۰	۰٫۰۰۰۱	۹۶	۷۷٫۵
۲۰۰	۰٫۰۰۰۱	۸۲	۷۷٫۷

۲۰۰	۰,۰۰۰۰۵	۷۲	۷۴,۲
۲۰۰	۰,۰۰۰۷	۲۳	**۷۸,۶**

۵-۶-۱-۴- نتایج شبکه با ویژگی‌های DBN

به‌منظور قابل مقایسه بودن نتایج DBN با نتایج MFCC، شبکه LSTM با پارامترهای مرتبط با بهترین نتیجه حاصل از ویژگی‌های MFCC آموزش داده شده است. همان‌طور که در جدول ۵-۴ دیده می‌شود، آموزش شبکه با ویژگی‌های حاصل از DBN دقت شبکه را به میزان ۱,۵٪ در مقایسه با ویژگی‌های MFCC افزایش داده است.

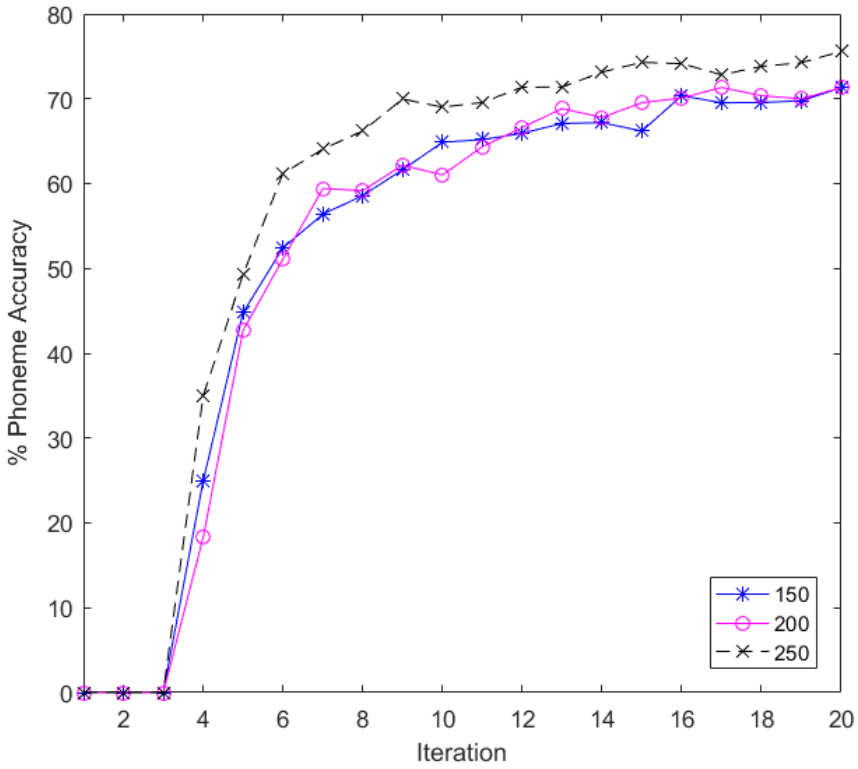
جدول ۵-۴) مقایسه نتایج دقت LSTM یک‌طرفه در سطح فریم با ویژگی‌های MFCC و DBN

روش استخراج ویژگی	تعداد بلوک حافظه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
MFCC	۲۰۰	۰,۰۰۰۷	۲۳	۷۸,۶
DBN	۲۰۰	۰,۰۰۰۷	۳۶	**۸۰,۱**

۵-۶-۲- تشخیص واج

۵-۶-۲-۱- تاثیر تعداد بلوک‌های حافظه

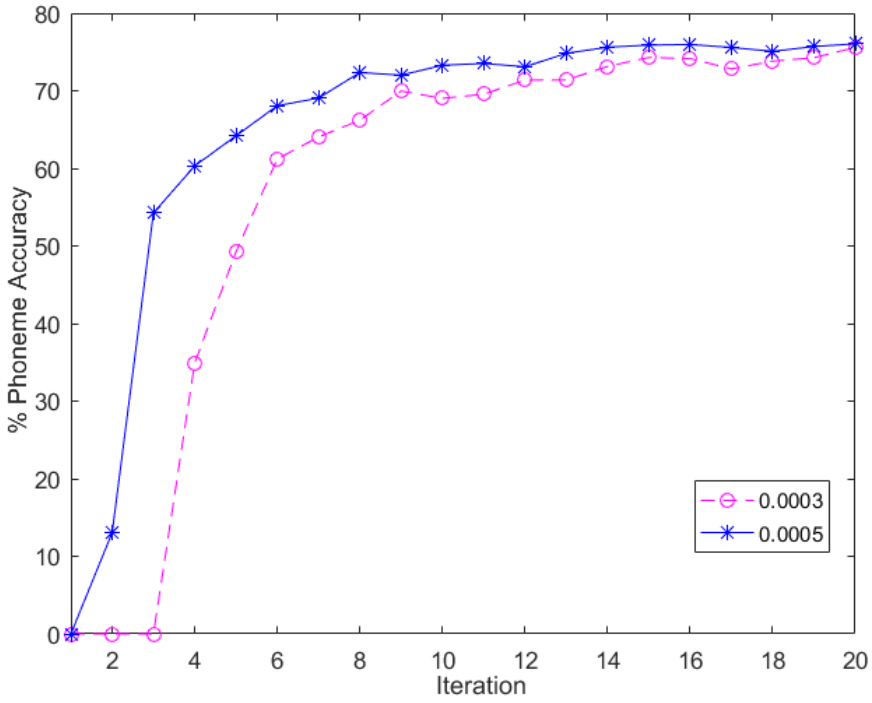
برای بررسی تاثیر این پارامتر، مقدار نرخ یادگیری برابر مقدار ثابت ۰,۰۰۰۳ در نظر گرفته شده است و شبکه به‌ازای ۱۵۰، ۲۰۰، ۲۵۰ بلوک حافظه ۲۰ مرحله آموزش داده شده است. همان‌طور که در شکل ۵-۳ دیده می‌شود، بهترین نتیجه مربوط به ۲۵۰ بلوک لایه میانی است.



شکل ۵-۳) دقت تشخیص واج LSTM به‌ازای نرخ یادگیری ۰,۰۰۰۳ در ۲۰ مرحله آموزش

۵-۶-۲-۲- تاثیر نرخ یادگیری

به‌منظور بررسی اثر نرخ یادگیری، تعداد بلوک‌های لایه پنهان برابر ۲۵۰ قرار داده شده است و شبکه به‌ازای چند نرخ یادگیری مختلف ۲۰ مرحله روی مجموعه فارس‌دات آموزش داده شده است. همان‌طور که در شکل ۵-۴ دیده می‌شود، افزایش نرخ یادگیری تاثیر مثبتی بر بهبود یادگیری شبکه ندارد و تنها سرعت یادگیری افزایش پیدا کرده است.



شکل ۵-۴) دقت تشخیص واج LSTM به‌ازای ۲۵۰ بلوک حافظه در ۲۰ مرحله آموزش

۵-۶-۲-۳- نتایج شبکه با ویژگی‌های MFCC

جدول ۵-۵ دقت شبکه را روی داده‌های تست به‌ازای پارامترهای مختلف نمایش می‌دهد. همان‌طور که دیده می‌شود، بهترین دقت مربوط به شبکه LSTM با ۲۵۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۳ می‌باشد که این دقت معادل ۷۷٪ است.

جدول ۵-۵) نتایج دقت LSTM یک‌طرفه در سطح واج روی داده‌های تست

تعداد بلوک حافظه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
۱۵۰	۰,۰۰۰۳	۳۶	۷۳,۶
۲۰۰	۰,۰۰۰۳	۴۵	۷۴,۲
۲۵۰	۰,۰۰۰۳	۵۶	**۷۷**
۲۵۰	۰,۰۰۰۵	۴۲	۷۶,۹

۵-۶-۲-۴- نتایج شبکه با ویژگی‌های DBN

به‌منظور قابل مقایسه بودن نتایج DBN با نتایج MFCC، شبکه LSTM با پارامترهای مرتبط با بهترین نتیجه حاصل از ویژگی‌های MFCC آموزش داده شده است. همان‌طور که در جدول ۵-۶ دیده می‌شود، آموزش شبکه با ویژگی‌های حاصل از DBN دقت شبکه را به میزان ۱٪ در مقایسه با ویژگی‌های MFCC افزایش داده است.

جدول ۵-۶) مقایسه نتایج دقت LSTM یک‌طرفه در سطح واج با ویژگی‌های MFCC و DBN

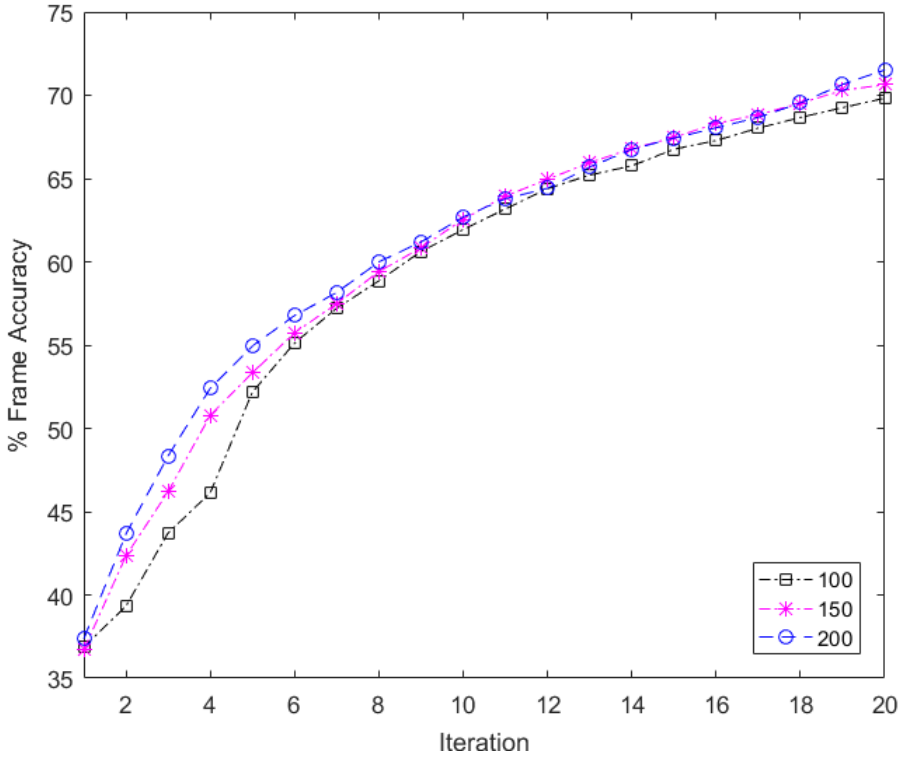
روش استخراج ویژگی	تعداد بلوک حافظه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
MFCC	۲۵۰	۰,۰۰۰۳	۵۶	۷۷
DBN	۲۵۰	۰,۰۰۰۳	۵۰	**۷۸**

۵-۷- نتایج شبکه عصبی حافظه کوتاه مدت ماندگار دوطرفه

۵-۷-۱- تشخیص فریم

۵-۷-۱-۱- تاثیر تعداد بلوک‌های حافظه

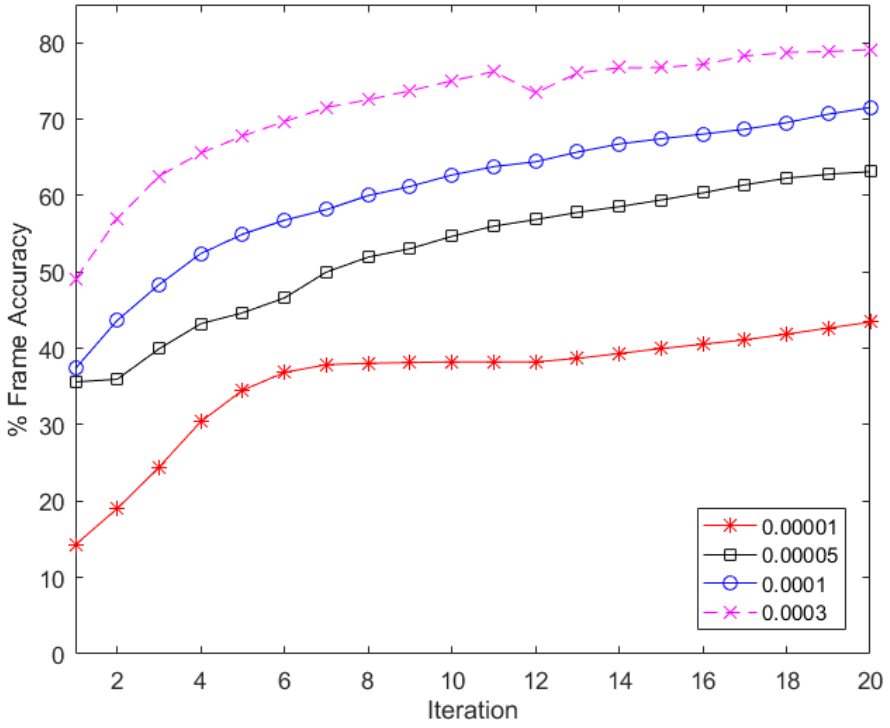
برای بررسی این پارامتر، مقدار نرخ یادگیری برابر مقدار ثابت ۰,۰۰۰۱ در نظر گرفته شده است و شبکه به‌ازای ۱۰۰، ۱۵۰ و ۲۰۰ بلوک لایه میانی ۲۰ مرحله آموزش داده شده است. همان‌طور که در شکل ۵-۵ دیده می‌شود، بهترین نتیجه با ۲۰۰ بلوک لایه میانی حاصل شده است.



شکل ۵-۵) دقت تشخیص فریم BLSTM به‌ازای نرخ یادگیری ۰٫۰۰۰۱ در ۲۰ مرحله آموزش

۵-۷-۱-۲- تاثیر نرخ یادگیری

به‌منظور بررسی نرخ یادگیری، تعداد بلوک‌های لایه پنهان برابر ۲۰۰ قرار داده شده است و شبکه به‌ازای چند نرخ یادگیری مختلف ۲۰ مرحله روی مجموعه فارس‌دات آموزش داده شده است. همان‌طور که در شکل ۵-۶ دیده می‌شود، بهترین نتیجه مربوط به حالتی می‌باشد که نرخ یادگیری برابر ۰٫۰۰۰۳ است.



شکل ۵-۶) دقت تشخیص فریم BLSTM به‌ازای ۲۰۰ بلوک حافظه در ۲۰ مرحله آموزش

۵-۷-۱-۳- نتایج شبکه با ویژگی‌های MFCC

جدول ۵-۷ دقت تشخیص فریم شبکه BLSTM را روی داده‌های تست به‌ازای پارامترهای مختلف نمایش می‌دهد. همان‌طور که دیده می‌شود، بهترین دقت مربوط به شبکه با ۲۰۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۳ می‌باشد که این دقت معادل ۸۱,۵٪ است و نسبت به حالت یک‌طرفه به‌میزان ۲,۹٪ افزایش دقت داشته است.

جدول ۵-۷) نتایج دقت BLSTM در سطح فریم روی داده‌های تست

تعداد بلوک حافظه در هر لایه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
۱۰۰	۰,۰۰۰۱	۱۰۰	۸۰,۸
۱۵۰	۰,۰۰۰۱	۱۰۱	۸۰,۱
۲۰۰	۰,۰۰۰۱	۱۰۰	۸۱,۲
۲۰۰	۰,۰۰۰۳	۳۷	**۸۱,۵**
۲۰۰	۰,۰۰۰۰۱	۱۰۱	۶۴

۲۰۰	۰,۰۰۰۰۵	۹۹	۷۹,۴

۵-۷-۱-۴- نتایج شبکه با ویژگی‌های DBN

به‌منظور قابل مقایسه بودن نتایج DBN با نتایج MFCC، شبکه BLSTM با پارامترهای مرتبط با بهترین نتیجه حاصل از ویژگی‌های MFCC آموزش داده شده است. همان‌طور که در جدول ۵-۸ دیده می‌شود، آموزش شبکه با ویژگی‌های حاصل از DBN دقت شبکه را به میزان ۰,۹٪ در مقایسه با ویژگی‌های MFCC افزایش داده است.

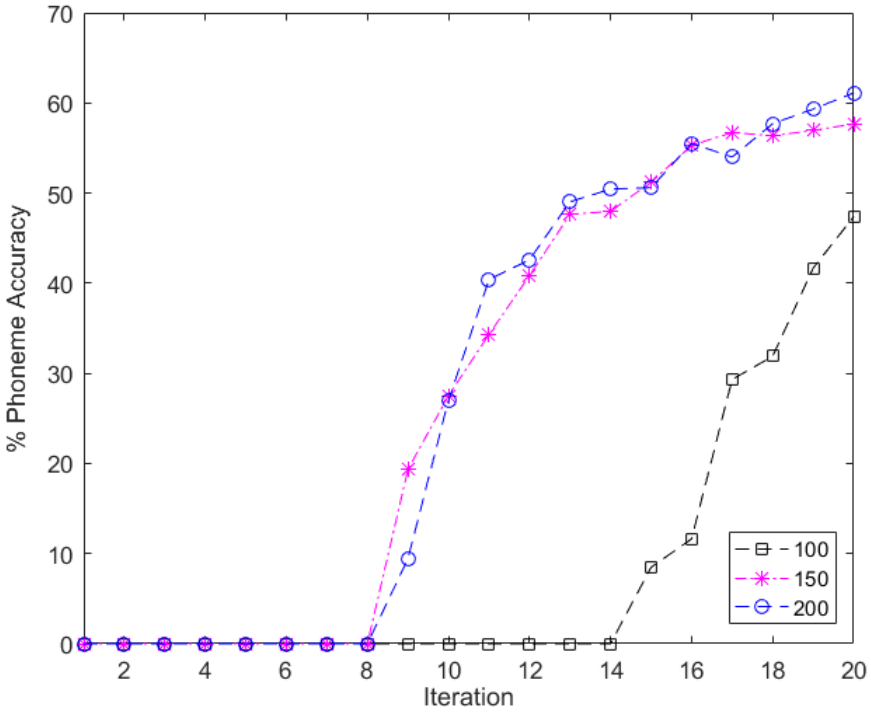
جدول ۵-۸) مقایسه نتایج دقت BLSTM در سطح فریم با ویژگی‌های MFCC و DBN

روش استخراج ویژگی	تعداد بلوک حافظه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
MFCC	۲۰۰	۰,۰۰۰۳	۳۷	۸۱,۵
DBN	۲۰۰	۰,۰۰۰۳	۸۱	**۸۲,۴**

۵-۷-۲- تشخیص واج

۵-۷-۲-۱- تاثیر تعداد بلوک‌های حافظه

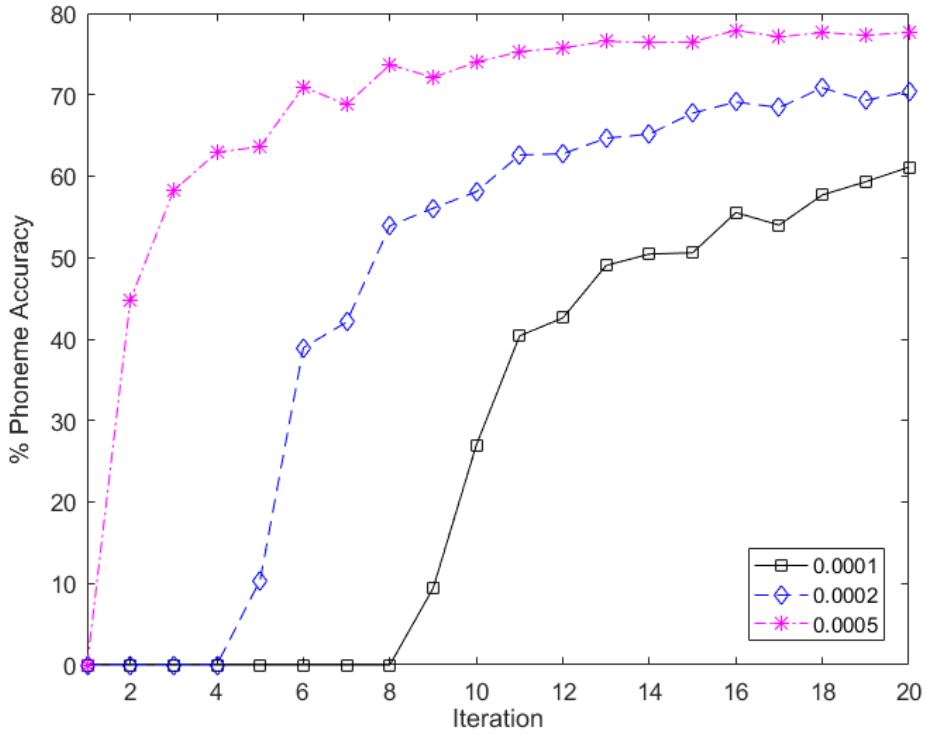
جهت بررسی تاثیر این پارامتر، مقدار نرخ یادگیری برابر مقدار ثابت ۰,۰۰۰۱ در نظر گرفته شده است و شبکه به‌ازای۱۰۰، ۱۵۰ و ۲۰۰ بلوک لایه میانی ۲۰ مرحله آموزش داده شده است. همان‌طور که در شکل ۵-۷ دیده می‌شود، بهترین نتیجه با ۲۰۰ بلوک لایه میانی حاصل شده است.



شکل ۵-۷) دقت تشخیص واج BLSTM به‌ازای نرخ یادگیری ۰٫۰۰۰۱ در ۲۰ مرحله آموزش

۵-۷-۲-۲- تاثیر نرخ یادگیری

به‌منظور بررسی تاثیر نرخ یادگیری، تعداد بلوک‌های لایه پنهان برابر ۲۰۰ قرار داده شده است و شبکه به‌ازای چند نرخ یادگیری مختلف ۲۰ مرحله روی مجموعه فارس‌دات آموزش داده شده است. همان‌طور که در شکل ۵-۸ دیده می‌شود، بهترین نتیجه مربوط به حالتی می‌باشد که نرخ یادگیری برابر ۰٫۰۰۰۵ است.



شکل ۵-۸) دقت تشخیص واج BLSTM به‌ازای ۲۰۰ بلوک حافظه در ۲۰ مرحله آموزش

۵-۷-۲-۳- نتایج شبکه با ویژگی‌های MFCC

جدول ۵-۹ دقت تشخیص واج شبکه BLSTM را روی داده‌های تست به‌ازای پارامترهای مختلف نمایش می‌دهد. همان‌طور که دیده می‌شود، بهترین دقت مربوط به شبکه با ۲۰۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۵ می‌باشد که این دقت معادل ۷۹,۳٪ است و نسبت به حالت یک‌طرفه به‌میزان ۲,۳٪ افزایش دقت داشته است.

جدول ۵-۹) نتایج دقت BLSTM در سطح واج روی داده‌های تست

تعداد بلوک حافظه در هر لایه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
۱۰۰	۰,۰۰۰۱	۴۸	۶۹,۳
۱۵۰	۰,۰۰۰۱	۴۵	۷۲
۲۰۰	۰,۰۰۰۱	۵۱	۷۳,۸
۲۰۰	۰,۰۰۰۲	۳۳	۷۵
۲۰۰	۰,۰۰۰۵	۳۲	**۷۹,۳**

۵-۷-۲-۴- نتایج شبکه با ویژگی‌های DBN

به‌منظور قابل مقایسه بودن نتایج DBN با نتایج MFCC، شبکه BLSTM با پارامترهای مرتبط با بهترین نتیجه حاصل از ویژگی‌های MFCC آموزش داده شده است. همان‌طور که در جدول ۵-۱۰ نیز دیده می‌شود، آموزش شبکه با ویژگی‌های حاصل از DBN به میزان ۱٫۱٪ دقت کمتری در مقایسه با ویژگی‌های MFCC داشته است.

جدول ۵-۱۰) مقایسه نتایج دقت BLSTM در سطح واج با ویژگی‌های MFCC و DBN

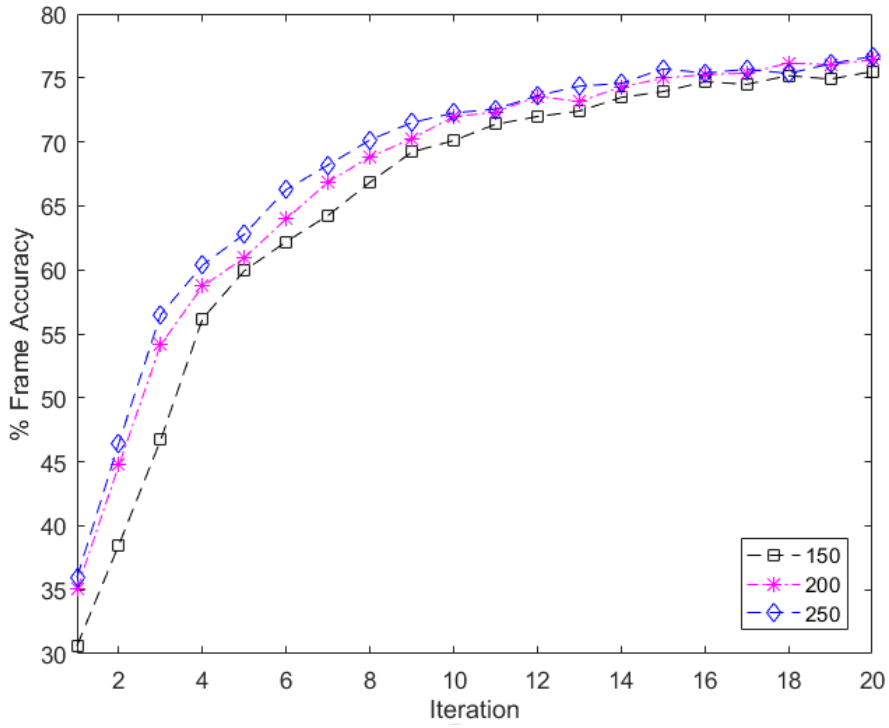
روش استخراج ویژگی	تعداد بلوک حافظه	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
MFCC	۲۰۰	۰٫۰۰۰۵	۳۲	**۷۹٫۳**
DBN	۲۰۰	۰٫۰۰۰۵	۴۴	۷۸٫۲

۵-۸- نتایج شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه

۵-۸-۱- تشخیص فریم

۵-۸-۱-۱- تاثیر تعداد بلوک‌های حافظه

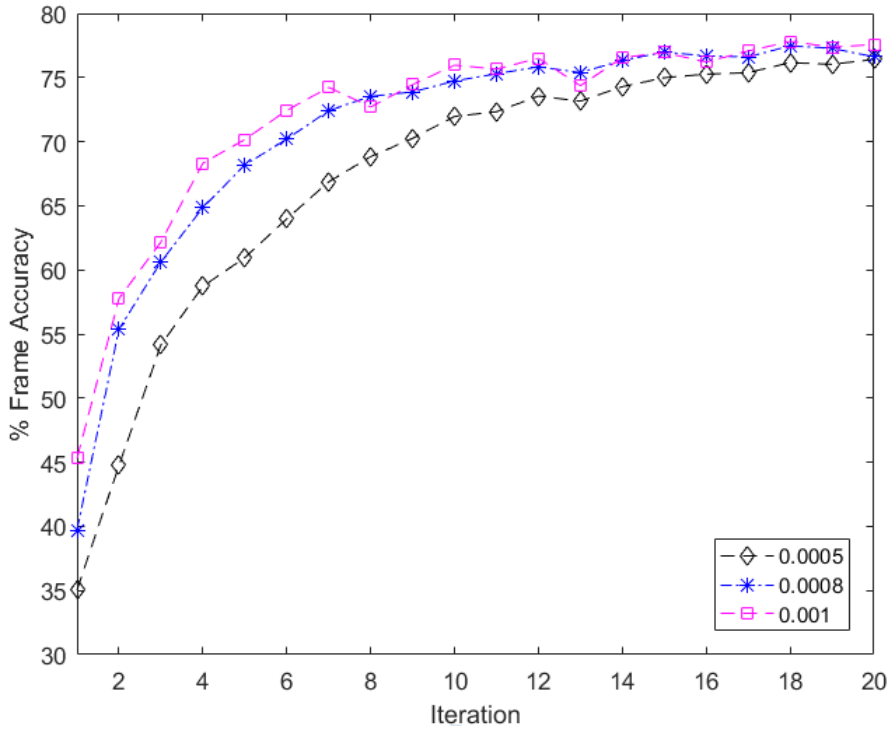
برای بررسی این پارامتر، مقدار نرخ یادگیری برابر مقدار ثابت ۰٫۰۰۰۵ در نظر گرفته شده است و شبکه به‌ازای ۱۵۰، ۲۰۰ و ۲۵۰ بلوک لایه میانی و تعداد ۲ لایه پنهان در ۲۰ مرحله آموزش داده شده است. همان‌طور که در شکل ۵-۹ دیده می‌شود، افزایش تعداد بلوک‌های حافظه تاثیر چندانی بر افزایش قدرت یادگیری شبکه ندارد.



شکل ۵-۹) دقت تشخیص فریم DLSTM به‌ازای نرخ یادگیری ۰٫۰۰۰۵ و ۲ لایه پنهان در ۲۰ مرحله آموزش

۵-۸-۱-۲- تاثیر نرخ یادگیری

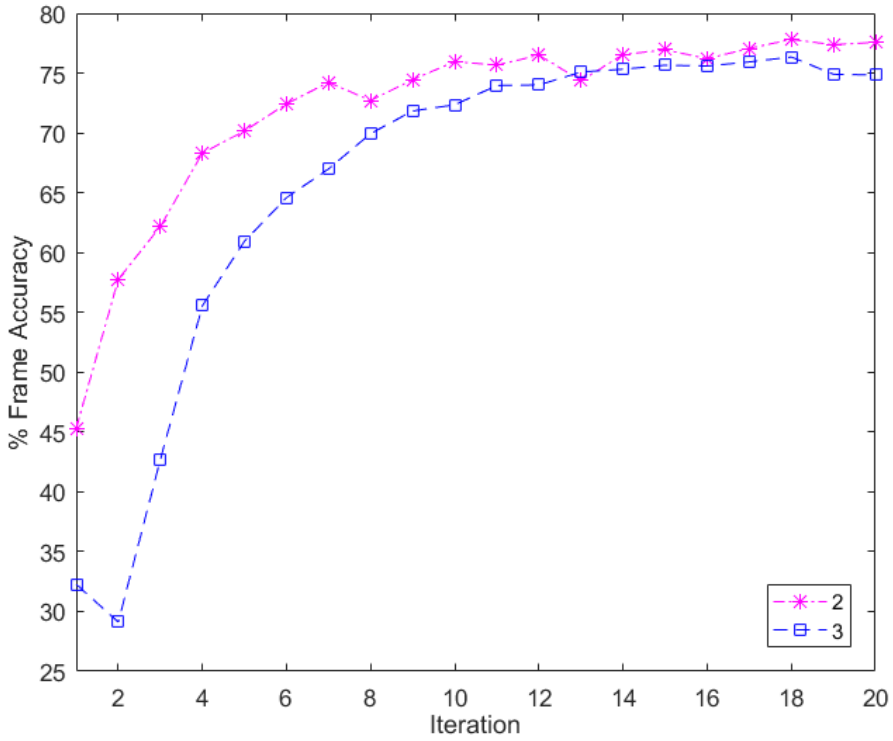
جهت بررسی تاثیر نرخ یادگیری، تعداد بلوک‌های لایه پنهان برابر ۲۰۰ قرار داده شده است و تعداد لایه‌های پنهان نیز ۲ در نظر گرفته شده است. سپس شبکه به‌ازای چند نرخ یادگیری مختلف ۲۰ مرحله روی مجموعه فارس‌دات آموزش داده شده است. همان‌طور که در شکل ۵-۱۰ دیده می‌شود، بهترین نتیجه مربوط به حالتی می‌باشد که نرخ یادگیری برابر ۰٫۰۰۱ است اگرچه افزایش نرخ یادگیری تاثیر زیادی بر سرعت یادگیری شبکه نداشته است.



شکل ۵-۱۰) دقت تشخیص فریم DLSTM به‌ازای ۲۰۰ بلوک حافظه و ۲ لایه پنهان در ۲۰ مرحله آموزش

۵-۸-۱-۳- تاثیر تعداد لایه‌های پنهان

شکل ۵-۱۱ تاثیر افزایش تعداد لایه‌ها را برای ۲۰۰ بلوک حافظه و نرخ یادگیری برابر مقدار ثابت ۰,۰۰۱ نمایش می‌دهد. همان‌طور که دیده می‌شود، با افزایش تعداد لایه‌ها به‌دلیل زیاد شدن تعداد پارامترهای شبکه یادگیری شبکه کندتر انجام می‌شود.



شکل ۵-۱۱) دقت تشخیص فریم DLSTM به‌ازای ۲۰۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۱ در ۲۰ مرحله آموزش

۵-۸-۱-۴- نتایج شبکه با ویژگی‌های MFCC

جدول ۵-۱۱ دقت تشخیص فریم شبکه DLSTM را روی داده‌های تست به‌ازای پارامترهای مختلف نمایش می‌دهد. بهترین دقت مربوط به شبکه با ۲۰۰ بلوک حافظه، ۲ لایه پنهان و نرخ یادگیری ۰,۰۰۱ می‌باشد که این دقت معادل ۸۰,۲٪ است و نسبت به حالت یک لایه به‌میزان ۱,۶٪ افزایش دقت داشته است.

جدول ۵-۱۱) نتایج دقت DLSTM در سطح فریم روی داده‌های تست

تعداد بلوک حافظه در هر لایه	تعداد لایه‌ها	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
۱۵۰	۲	۰,۰۰۰۵	۳۹	۷۹,۴
۲۰۰	۲	۰,۰۰۰۵	۲۸	۷۸,۷
۲۵۰	۲	۰,۰۰۰۵	۴۱	۷۹,۷
۲۰۰	۲	۰,۰۰۰۸	۴۳	۷۹,۸
۲۰۰	۲	۰,۰۰۱	۳۴	**۸۰,۲**

۷۹,۵	۳۱	۰,۰۰۱	۳	۲۰۰

۵-۸-۱-۵- نتایج شبکه با ویژگی‌های DBN

به‌منظور قابل مقایسه بودن نتایج DBN با نتایج MFCC، شبکه DLSTM با پارامترهای مرتبط با بهترین نتیجه حاصل از ویژگی‌های MFCC آموزش داده شده است. جدول ۵-۱۲ خلاصه نتایج به‌دست آمده را نمایش می‌دهد. با توجه به نتایج به‌دست آمده، آموزش شبکه با ویژگی‌های حاصل از DBN دقت شبکه را ۱٪ در مقایسه با ویژگی‌های MFCC افزایش داده است.

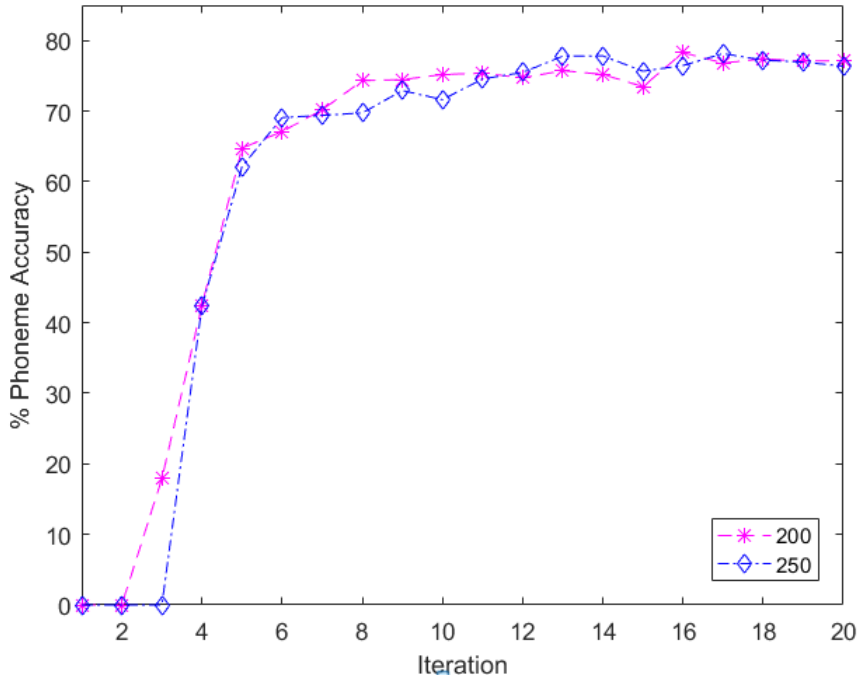
جدول ۵-۱۲) مقایسه نتایج دقت DLSTM در سطح فریم با ویژگی‌های MFCC و DBN

روش استخراج ویژگی	تعداد بلوک حافظه	تعداد لایه‌ها	نرخ یادگیری	تعداد مراحل آموزش	دقت تست
MFCC	۲۰۰	۲	۰,۰۰۱	۳۴	۸۰,۲
DBN	۲۰۰	۲	۰,۰۰۱	۲۹	**۸۱,۲**

۵-۸-۲- تشخیص واج

۵-۸-۲-۱- تاثیر تعداد بلوک‌های حافظه

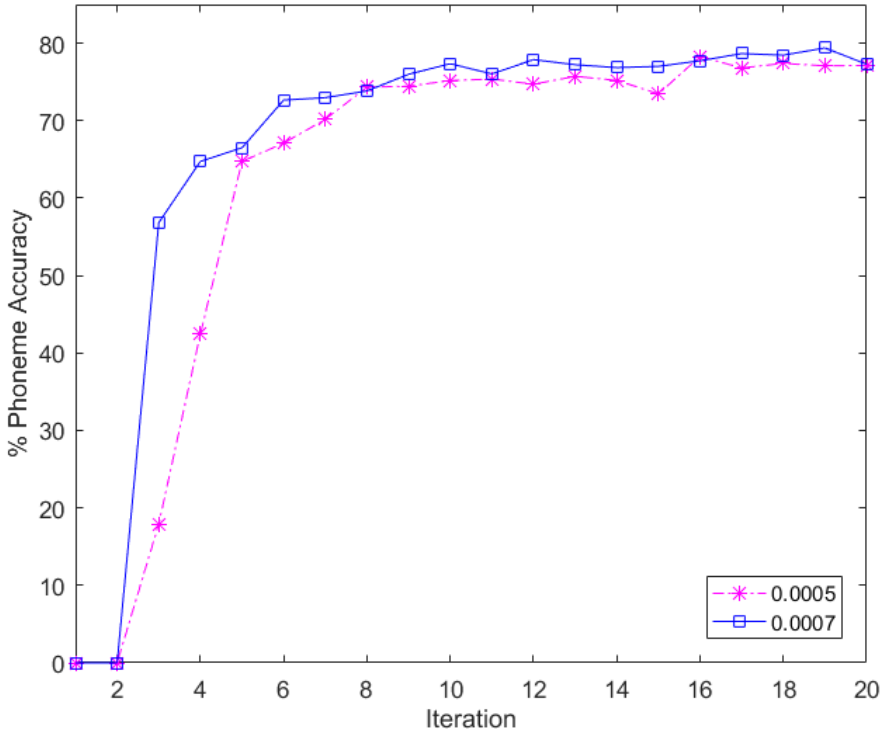
برای بررسی این پارامتر، مقدار نرخ یادگیری برابر مقدار ثابت ۰,۰۰۰۵ در نظر گرفته شده است و شبکه به‌ازای ۲۰۰ و ۲۵۰ بلوک لایه میانی و تعداد ۲ لایه پنهان در ۲۰ مرحله آموزش داده شده است. همان‌طور که در شکل ۵-۱۲ دیده می‌شود، افزایش تعداد بلوک‌های حافظه تاثیر قابل توجهی بر افزایش قدرت یادگیری شبکه ندارد.



شکل ۵-۱۲) دقت تشخیص واج DLSTM به‌ازای نرخ یادگیری ۰،۰۰۰۵ و ۲ لایه پنهان در ۲۰ مرحله آموزش

۵-۸-۲-۲- تاثیر نرخ یادگیری

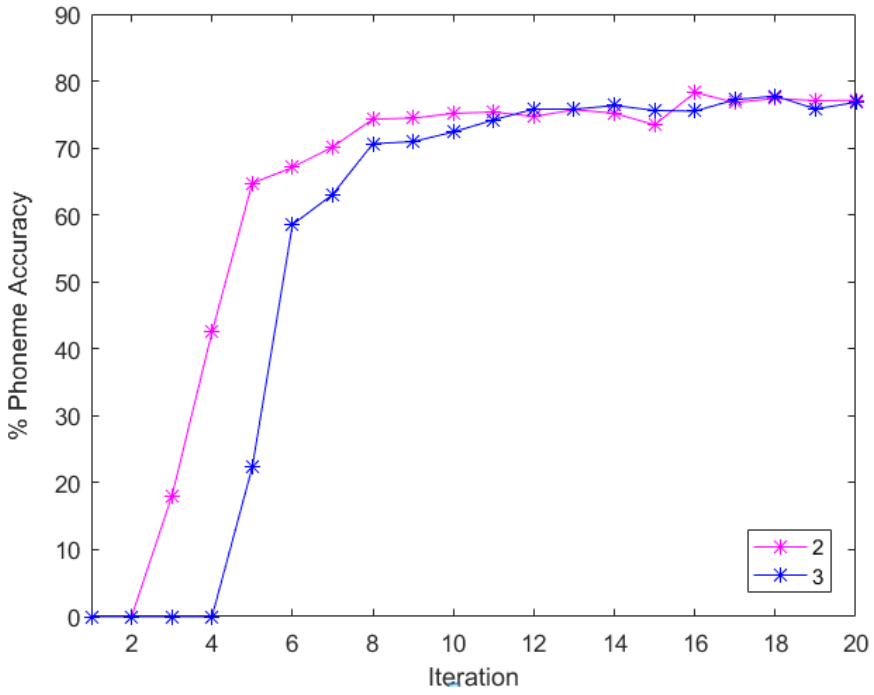
جهت بررسی تاثیر نرخ یادگیری، تعداد بلوک‌های لایه پنهان برابر ۲۰۰ قرار داده شده است و تعداد لایه‌های پنهان ۲ در نظر گرفته شده است. سپس شبکه به‌ازای چند نرخ یادگیری مختلف ۲۰ مرحله روی مجموعه فارس‌دات آموزش داده شده است. همان‌طور که در شکل ۵-۱۳ دیده می‌شود، افزایش نرخ یادگیری تاثیر قابل توجهی بر سرعت یادگیری شبکه نداشته است.



شکل ۵-۱۳) دقت تشخیص واج DLSTM به‌ازای ۲۰۰ بلوک حافظه و ۲ لایه پنهان در ۲۰ مرحله آموزش

۵-۸-۲-۳- تاثیر تعداد لایه‌های پنهان

شکل ۵-۱۴ تاثیر افزایش تعداد لایه‌ها را به‌ازای۲۰۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۵ نمایش می‌دهد. همان‌طور که دیده می‌شود افزایش تعداد لایه روند یادگیری را کندتر کرده است.



شکل ۵-۱۴) دقت تشخیص واج DLSTM به‌ازای ۲۰۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۵ در ۲۰ مرحله آموزش

۵-۸-۲-۴- نتایج شبکه با ویژگی‌های MFCC

جدول ۵-۱۳ دقت تشخیص واج شبکه DLSTM را روی داده‌های تست به‌ازای پارامترهای مختلف نمایش می‌دهد. بهترین دقت مربوط به شبکه با ۲۰۰ بلوک حافظه، ۲ لایه پنهان و نرخ یادگیری ۰,۰۰۰۵ می‌باشد و معادل ۸۰,۳٪ است و نسبت به حالت یک لایه به‌میزان ۳,۳٪ افزایش دقت داشته است.

جدول ۵-۱۳) نتایج دقت DLSTM در سطح واج روی داده‌های تست

تعداد بلوک حافظه در هر لایه	تعداد لایه‌ها	نرخ یادگیری	تعداد مراحل آموزش	دقت داده‌های تست
۲	۲۰۰	۰,۰۰۰۵	۵۳	**۸۰,۳**
۲	۲۵۰	۰,۰۰۰۵	۳۸	۸۰,۲
۲	۲۰۰	۰,۰۰۰۷	۲۹	۷۸,۷
۳	۲۰۰	۰,۰۰۰۵	۵۱	۸۰,۱

۵-۸-۲-۵- نتایج شبکه با ویژگی‌های DBN

به‌منظور قابل مقایسه بودن نتایج DBN با نتایج MFCC، شبکه DLSTM با پارامترهای مرتبط با بهترین نتیجه حاصل از ویژگی‌های MFCC آموزش داده شده است. جدول ۵-۱۴ خلاصه نتایج به‌دست آمده را نمایش می‌دهد. با توجه به نتایج به‌دست آمده، آموزش شبکه با ویژگی‌های حاصل از DBN دقت شبکه را در حدود ۱٫۸٪ در مقایسه با ویژگی‌های MFCC کاهش داده است.

جدول ۵-۱۴) مقایسه نتایج دقت DLSTM در سطح واج با ویژگی‌های MFCC و DBN

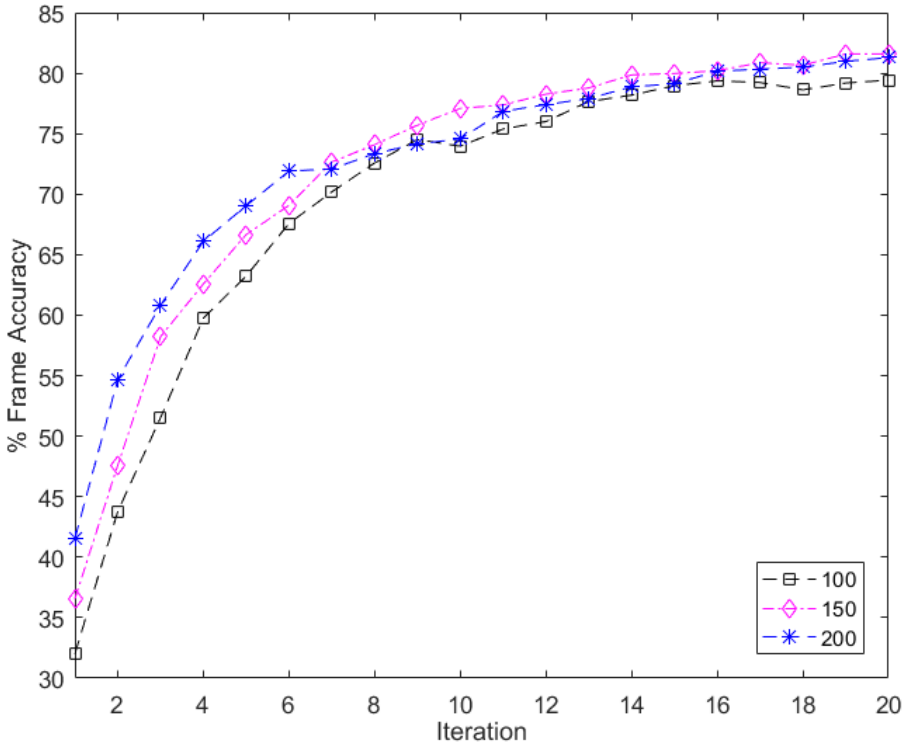
روش استخراج ویژگی	تعداد بلوک حافظه	تعداد لایه‌ها	نرخ یادگیری	تعداد مراحل آموزش	دقت تست
MFCC	۲۰۰	۲	۰٫۰۰۰۵	۵۳	**۸۰٫۳**
DBN	۲۰۰	۲	۰٫۰۰۰۵	۲۹	۷۸٫۵

۵-۹- نتایج شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه

۵-۹-۱- تشخیص فریم

۵-۹-۱-۱- تاثیر تعداد بلوک‌های حافظه

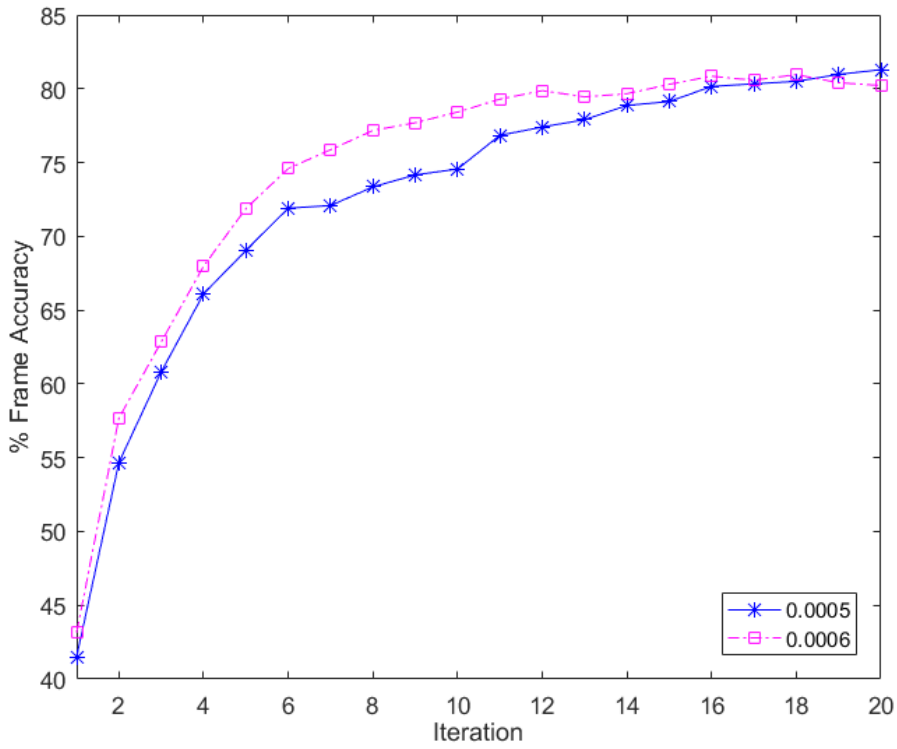
برای بررسی این پارامتر، مقدار نرخ یادگیری برابر مقدار ثابت ۰٫۰۰۰۵ و تعداد لایه‌های پنهان برابر ۲ قرار داده شده و شبکه به‌ازای ۱۰۰، ۱۵۰ و ۲۰۰ بلوک حافظه در ۲۰ مرحله آموزش داده شده است. همان‌طور که در شکل ۵-۱۵ دیده می‌شود، دقت شبکه به‌ازای ۱۵۰ و ۲۰۰ بلوک حافظه بسیار به یکدیگر نزدیک است و این به بدین معنا می‌باشد که، افزایش بیشتر بلوک حافظه تاثیری بر افزایش دقت شبکه نخواهد داشت.



شکل ۵-۱۵) دقت تشخیص فریم DBLSTM به‌ازای نرخ یادگیری ۰،۰۰۰۵ و ۲ لایه پنهان در ۲۰ مرحله آموزش

۵-۹-۱-۲- تاثیر نرخ یادگیری

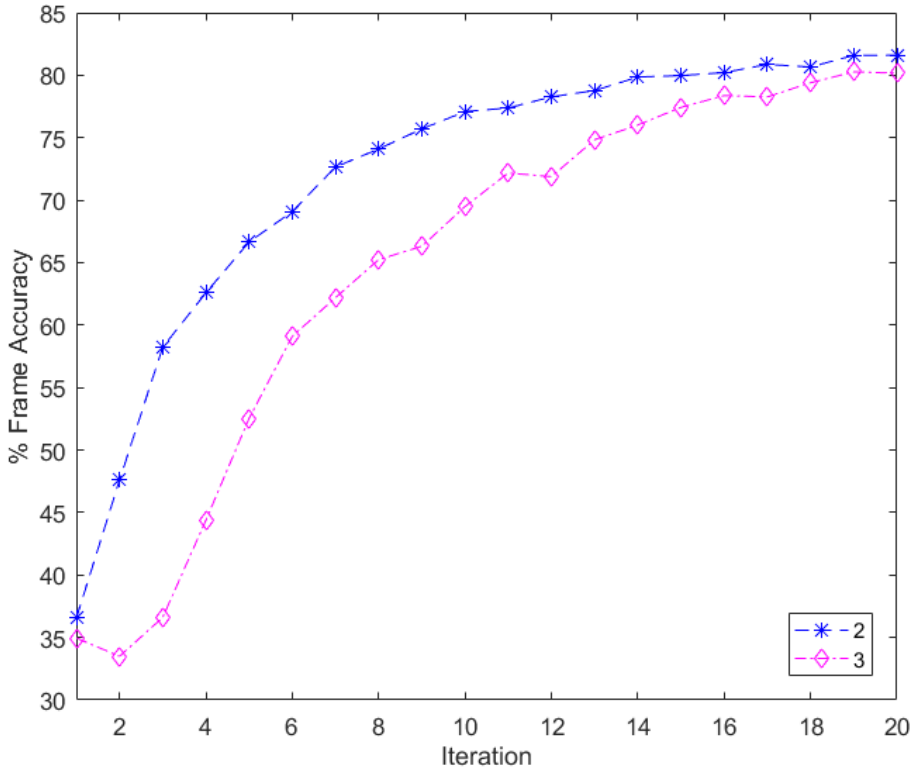
نمودار ۵-۱۶ تاثیر نرخ یادگیری بر روند آموزش شبکه نشان می‌دهد. برای این منظور تعداد لایه‌های پنهان برابر ۲ و تعداد بلوک حافظه ۲۰۰ در نظر گرفته شده است. همان‌طور که دیده می‌شود افزایش نرخ یادگیری روی بهبود روند آموزش تاثیر قابل توجهی ندارد.



شکل ۵-۱۶) دقت تشخیص فریم DBLSTM به‌ازای ۲۰۰ بلوک حافظه و ۲ لایه پنهان در ۲۰ مرحله آموزش

۵-۹-۱-۳- تاثیر تعداد لایه‌های پنهان

به منظور بررسی تاثیر افزایش تعداد لایه‌های پنهان بر یادگیری شبکه تعداد نرون‌های لایه های پنهان برابر مقدار ثابت ۱۵۰ قرار داده شده است و همچنین نرخ یادگیری معادل ۰,۰۰۰۵ در نظر گرفته شده است. شکل ۵-۱۷ تاثیر این پارامتر را نمایش می‌دهد. همان‌طور که دیده می‌شود با افزایش تعداد لایه‌ها، به دلیل افزایش تعداد پارامترهای مدل سرعت یادگیری کاهش یافته است.



شکل ۵-۱۷) دقت تشخیص فریم DBLSTM به‌ازای ۱۵۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۵ در ۲۰ مرحله آموزش

۵-۹-۱-۴- نتایج شبکه با ویژگی‌های MFCC

جدول ۵-۱۵ دقت تشخیص فریم شبکه DBLSTM را روی داده‌های تست به‌ازای پارامترهای مختلف نمایش می‌دهد. بهترین دقت به‌دست آمده مربوط به شبکه ۳ لایه با ۱۵۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۵ می‌باشد که این دقت معادل ۸۳,۸٪ است که نسبت به حالت یک لایه به‌میزان ۲,۳٪ افزایش دقت داشته است.

جدول ۵-۱۵) نتایج دقت DBLSTM در سطح فریم روی داده‌های تست

تعداد بلوک حافظه در هر لایه	نرخ یادگیری	تعداد لایه‌های پنهان	تعداد مراحل آموزش	دقت داده‌های تست
۱۵۰	۰,۰۰۰۵	۲	۴۶	۸۳,۴
۲۰۰	۰,۰۰۰۵	۲	۴۸	۸۳,۶
۱۰۰	۰,۰۰۰۵	۲	۳۲	۸۲,۷
۲۰۰	۰,۰۰۰۶	۲	۳۶	۸۳

۱۵۰	۰,۰۰۰۵	۳	۴۹	**۸۳,۸**

۵-۹-۱-۵- نتایج شبکه با ویژگی‌های DBN

به‌منظور قابل مقایسه بودن نتایج DBN با نتایج MFCC، شبکه DBLSTM با پارامترهای مرتبط با بهترین نتیجه حاصل از ویژگی‌های MFCC آموزش داده شده است. جدول ۵-۱۶ خلاصه نتایج به‌دست آمده را نمایش می‌دهد. با توجه به نتایج به‌دست آمده، آموزش شبکه با ویژگی‌های حاصل از DBN دقت شبکه را در حدود ۰,۳٪ در مقایسه با ویژگی‌های MFCC کاهش داده است.

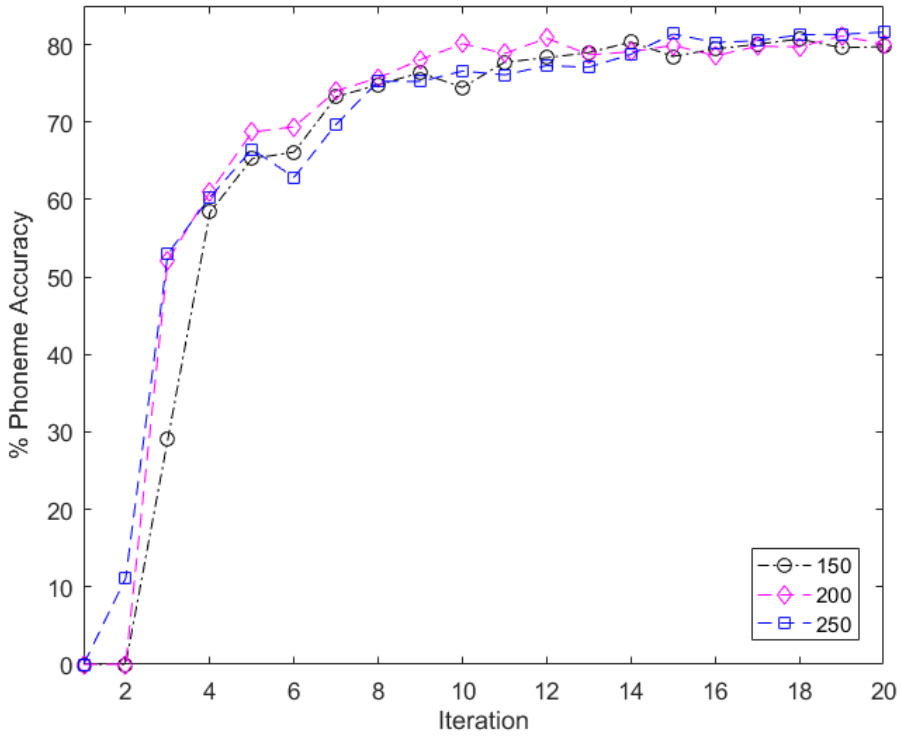
جدول ۵-۱۶) مقایسه نتایج دقت DBLSTM در سطح فریم با ویژگی‌های MFCC و DBN

روش استخراج ویژگی	تعداد بلوک حافظه	تعداد لایه‌ها	نرخ یادگیری	تعداد مراحل آموزش	دقت تست
MFCC	۱۵۰	۳	۰,۰۰۰۵	۴۹	**۸۳,۸**
DBN	۱۵۰	۳	۰,۰۰۰۵	۴۱	۸۳,۵

۵-۹-۲- تشخیص واج

۵-۹-۲-۱- تاثیر تعداد بلوک‌های حافظه

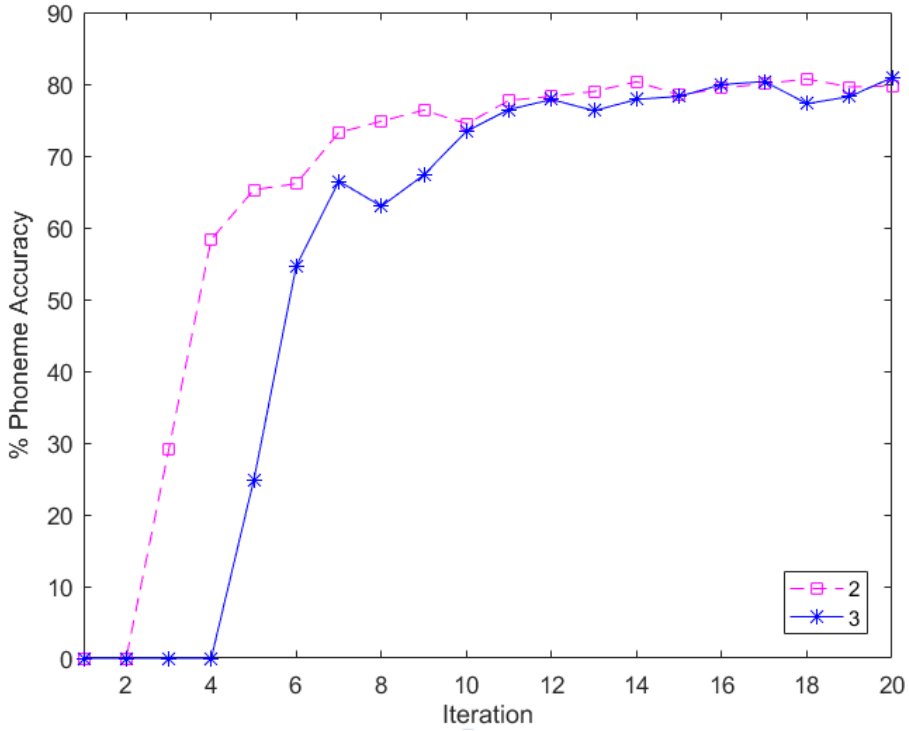
برای بررسی این پارامتر، مقدار نرخ یادگیری برابر مقدار ثابت ۰,۰۰۰۵ و تعداد لایه‌های پنهان برابر ۲ قرار داده شده است و شبکه به‌ازای ۱۵۰، ۲۰۰ و ۲۵۰ بلوک حافظه در ۲۰ مرحله آموزش داده شده است. همان‌طور که در شکل ۵-۱۸ دیده می‌شود، رفتار سه نمودار بسیار به یکدیگر شبیه است و افزایش تعداد بلوک حافظه تاثیر مثبتی بر افزایش دقت شبکه ندارد.



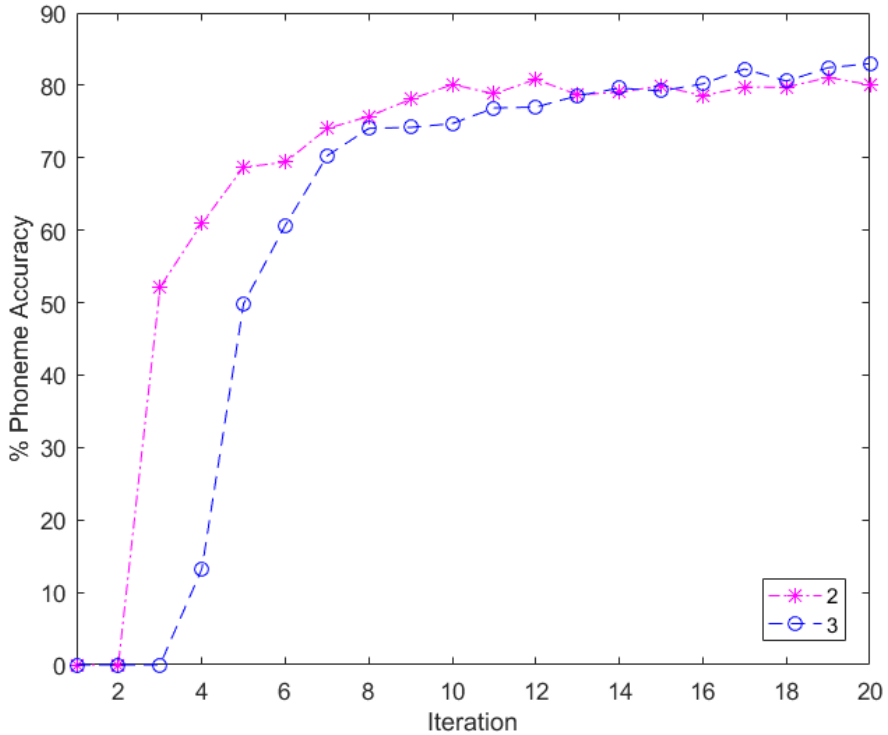
شکل ۵-۱۸) دقت تشخیص واج DBLSTM به‌ازای نرخ یادگیری ۰,۰۰۰۵ و ۲ لایه پنهان در ۲۰ مرحله آموزش

۵-۹-۲-۲- تاثیر تعداد لایه‌های پنهان

شکل‌های ۵-۱۹ و ۵-۲۰ به‌ترتیب تاثیر تعداد لایه‌های پنهان بر یادگیری شبکه به‌ازای ۱۵۰ و ۲۰۰ بلوک حافظه به‌ازای نرخ یادگیری ۰,۰۰۰۵ نشان می‌دهند. همان‌طور که دیده می‌شود با افزایش تعداد لایه‌ها، به دلیل افزایش پارامترهای مدل سرعت یادگیری کاهش یافته است.



شکل ۵-۱۹) دقت تشخیص واج DBLSTM به‌ازای ۱۵۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۵ در ۲۰ مرحله آموزش



شکل ۵-۲۰) دقت تشخیص واج DBLSTM به‌ازای ۲۰۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۵ در ۲۰ مرحله آموزش

۵-۹-۲-۳- نتایج شبکه با ویژگی‌های MFCC

جدول ۵-۱۷ دقت تشخیص واج شبکه DBLSTM را روی داده‌های تست به‌ازای پارامترهای مختلف نمایش می‌دهد. بهترین دقت به‌دست آمده مربوط به شبکه ۳ لایه با ۲۰۰ بلوک حافظه و نرخ یادگیری ۰,۰۰۰۵ می‌باشد. دقت به‌دست آمده معادل ۸۲,۹٪ است که نسبت به حالت یک لایه به‌میزان ۳,۶٪ افزایش دقت داشته است.

جدول ۵-۱۷) نتایج دقت DBLSTM در سطح واج روی داده‌های تست

تعداد بلوک حافظه در هر لایه	نرخ یادگیری	تعداد لایه‌های پنهان	تعداد مراحل آموزش	دقت داده‌های تست
۱۵۰	۰,۰۰۰۵	۲	۳۱	۸۱,۹
۲۰۰	۰,۰۰۰۵	۲	۴۹	۸۲,۷
۲۵۰	۰,۰۰۰۵	۲	۳۰	۸۱,۷
۲۰۰	۰,۰۰۰۵	۳	۳۱	**۸۲,۹**
۱۵۰	۰,۰۰۰۵	۳	۲۰	۷۹,۶

۵-۹-۲-۴- نتایج با ویژگی‌های DBN

به‌منظور قابل مقایسه بودن نتایج DBN با نتایج MFCC، شبکه DBLSTM با پارامترهای مرتبط با بهترین نتیجه حاصل از ویژگی‌های MFCC آموزش داده شده است. جدول ۵-۱۸ خلاصه نتایج به‌دست آمده را نمایش می‌دهد. با توجه به نتایج به‌دست آمده، آموزش شبکه با ویژگی‌های حاصل از DBN منجر به بهبود دقت شبکه به‌میزان ۰,۴٪ در مقایسه با ویژگی‌های MFCC شده است.

جدول ۵-۱۸) مقایسه نتایج دقت DBLSTM در سطح واج با ویژگی‌های MFCC و DBN

روش استخراج ویژگی	تعداد بلوک حافظه	تعداد لایه‌ها	نرخ یادگیری	تعداد مراحل آموزش	دقت تست
MFCC	۲۰۰	۳	۰,۰۰۰۵	۳۱	۸۲,۹
DBN	۲۰۰	۳	۰,۰۰۰۵	۳۷	**۸۳,۳**

۵-۱۰- مقایسه نتایج با مدل مخفی مارکوف

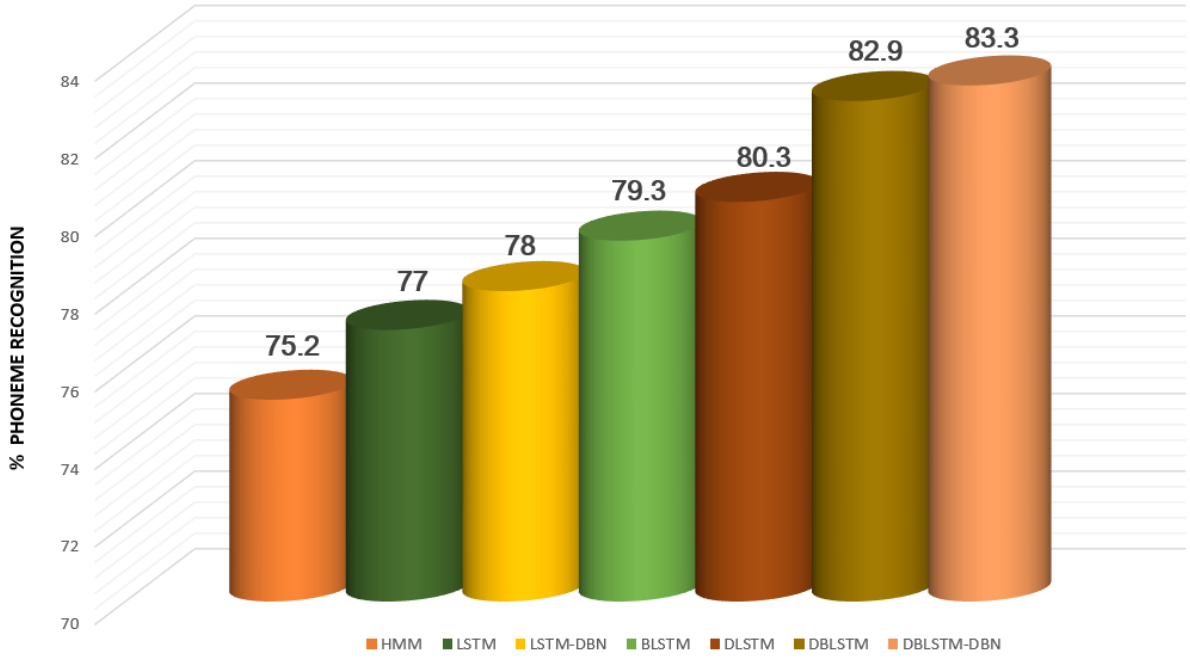
در این بخش نتایج به‌دست آمده از تشخیص واج با استفاده از با مدل مخفی مارکوف آورده شده است. به‌منظور دقیق بودن مقایسه، داده مورد استفاده جهت آموزش و آزمون مدل مخفی مارکوف و شبکه عصبی کاملا یکسان در نظر گرفته شده است. در مدل مخفی مارکوف مشابه شبکه عصبی ۳۹ ویژگی MFCC از هر فریم استخراج گردید. همچنین برای هر واج دو مدل ۳ حالته و ۵ حالته با ۸ و ۱۶ مدل مخلوط گاوسی[۸] (GMM) برای هر حالت ساخته شد. بهترین دقت روی مجموعه داده‌های تست در شرایطی حاصل گردید که هر واج با ۵ حالت مدل گردید و برای هر حالت ۱۶ مدل مخلوط گاوسی در نظر گرفته شد. نتایج به‌دست آمده برای تشخیص واج توسط مدل مخفی مارکوف در جدول ۵-۱۹ آورده شده است. همان‌طور که دیده می‌شود بهترین دقت روی مجموعه تست برابر ٪۷۵,۲ می‌باشد.

جدول ۵-۱۹) دقت تشخیص واج با مدل مخفی مارکوف

تعداد مدل مخلوط گاوسی	تعداد حالت	دقت داده آموزش	دقت داده تست
۱۶	۳	٪۷۵,۴	٪۷۲,۱
۱۶	۵	٪۷۸,۳	**٪۷۵,۲**
۸	۵	٪۷۵,۷	٪۷۳,۵
۸	۳	٪۷۱,۷	٪۶۹,۵

شکل ۵-۲۱ بهترین دقت تشخیص واج هر یک از روش‌های مدل‌سازی شده روی مجموعه فارس‌دات را نمایش می‌دهد. همان‌طور که دیده می‌شود بهترین دقت تشخیص واج با استفاده از شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه و ویژگی‌های شبکه باور عمیق حاصل گردیده است و معادل ٪۸۳,۳ می‌باشد که نسبت به مدل مخفی مارکوف به‌میزان ٪۸,۱ بهبود داشته است.

[۸] Gaussian Mixture Model (GMM)



شکل ۵-۲۱) بهترین دقت تشخیص واج برای هر یک از روش‌های مدل‌سازی

فصل ششم: جمع‌بندی و پیشنهاد برای آینده

۶-۱- خلاصه و جمع‌بندی

در این پایان‌نامه از شبکه عصبی حافظه کوتاه مدت ماندگار، شبکه عصبی حافظه کوتاه مدت ماندگار دوطرفه و همچنین شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه و دوطرفه جهت تشخیص واج مجموعه دادگان فارس‌دات استفاده گردید.

مجموعه فارس‌دات شامل ۶۰۸۰ سیگنال صوتی با نرخ نمونه برداری ۲۰ کیلو هرتز می‌باشد. این مجموعه شامل ۳۸۶ جمله است و توسط ۳۰۰ فارسی زبان که این افراد دارای ده لهجه متفاوت هستند خوانده شده است.

به‌منظور استخراج ویژگی ابتدا هر سیگنال به تعدادی فریم به طول ۱۶ میلی ثانیه و میزان هم‌پوشانی ۸ میلی ثانیه تبدیل شد و سپس با استفاده از دو روش MFCC و شبکه باور عمیق از هر فریم تعداد ۳۹ ویژگی استخراج گردید.

شبکه عصبی حافظه کوتاه مدت ماندگار یک شبکه عصبی بازگشتی است که در آن نرون‌های لایه پنهان با بلوک‌های حافظه جایگزین شده‌اند. استفاده از بلوک‌های حافظه در این شبکه موجب شده است تا مشکل فراموشی شبکه‌های عصبی بازگشتی برطرف گردد. بنابراین این شبکه برای پردازش دنباله‌های متوالی مناسب می‌باشد. شبکه عصبی حافظه کوتاه مدت ماندگار دوطرفه، شامل دو لایه پنهان مجزا با نام‌های پیش‌رو و رو به عقب می‌باشند که در آن‌ها نرون‌های این دو لایه پنهان با بلوک‌های حافظه LSTM جایگزین شده است. در این شبکه دنباله ورودی در دو جهت زمانی مخالف به این دو لایه پنهان بازگشتی داده می‌شود. بنابراین خروجی شبکه در هر گام زمانی به کل دنباله ورودی وابسته خواهد بود. در صورتی‌که، در شبکه حافظه کوتاه مدت ماندگار یک‌طرفه خروجی شبکه در هر مرحله تنها به ورودی فعلی و ورودی‌های پیشین وابسته می‌باشد. در صورتی‌که شبکه شامل چند لایه پنهان باشد شبکه عصبی عمیق حاصل می‌گردد. بنابراین با

روی هم قرار دادن نرون‌های لایه پنهان شبکه حافظه کوتاه مدت ماندگار دوطرفه، شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه حاصل می‌گردد و اگر نرون‌های لایه پنهان شبکه حافظه کوتاه مدت ماندگار یک‌طرفه روی یکدیگر قرار بگیرند شبکه حاصل، شبکه عصبی عمیق حافظه کوتاه مدت ماندگار یک‌طرفه نامیده می‌شود.

به‌منظور ارزیابی شبکه، دقت شبکه در دو سطح فریم و واج تعیین گردیده است. جهت تعیین دقت شبکه در سطح فریم، نسبت تعداد فریم‌های صحیح تشخیص داده شده توسط شبکه به تعداد کل فریم‌ها محاسبه گردیده است. همچنین به‌منظور ارزیابی دقت شبکه در سطح واج، از الگوریتم طبقه‌بند زمانی پیوندگرا جهت آموزش شبکه استفاده شده است. الگوریتم CTC این امکان را به شبکه LSTM می‌دهد که به‌جای برچسب گذاری هر فریم، دنباله واج متناظر با دنباله ورودی را تولید کند.

جدول ۶-۱ خلاصه نتایج به‌دست آمده روی مجموعه داده‌های تست را نمایش می‌دهد. لازم به توضیح است که همان‌طور که در جدول نتایج هر شبکه نیز آمده است، بهترین نتایج تشخیص واج و فریم برای هر شبکه لزوما با ساختار و پارامترهای یکسان حاصل نشده است. همان‌طور که در جدول ۶-۱ دیده می‌شود، تشخیص واج با شبکه عصبی عمیق حافظه کوتاه مدت ماندگار دوطرفه با استفاده از ویژگی‌های حاصل از شبکه باور عمیق نسبت به مدل مخفی مارکوف ۸,۱٪ بهبود داشته است و استفاده از شبکه عصبی عمیق نسبت به شبکه عصبی یک لایه کارایی سیستم را در هر دو سطح تشخیص فریم و واج بهبود داده است. همچنین شبکه‌های عصبی دوطرفه در هر دو حالت عمیق و غیر عمیق جهت تشخیص فریم و واج کارایی بالاتری نسبت به شبکه‌های عصبی یک‌طرفه دارند. علاوه بر این استفاده از ویژگی‌های به‌دست آمده از شبکه باور عمیق منجر به افزایش دقت تشخیص واج شبکه عمیق حافظه کوتاه مدت ماندگار دوطرفه به‌میزان ۰,۴٪ و شبکه حافظه کوتاه مدت ماندگار یک‌طرفه غیر عمیق به‌میزان ۱٪ در مقایسه با ویژگی‌های MFCC گردیده است. همچنین استفاده از ویژگی‌های این شبکه، منجر به افزایش دقت تشخیص فریم شبکه حافظه کوتاه مدت ماندگار یک‌طرفه و دوطرفه غیر عمیق و شبکه عمیق حافظه کوتاه مدت ماندگار یک‌طرفه در حدود ۱٪ در مقایسه با ویژگی‌های MFCC شده است.

جدول ۶-۱) خلاصه نتایج به‌دست آمده روی مجموعه فارس‌دات

روش مدل‌سازی	روش استخراج ویژگی	دقت تشخیص فریم	دقت تشخیص واج
شبکه LSTM	MFCC	۷۸,۶	۷۷
	DBN	۸۰,۱	۷۸
شبکه BLSTM	MFCC	۸۱,۵	۷۹,۳
	DBN	۸۲,۴	۷۸,۲
شبکه DLTM	MFCC	۸۰,۲	۸۰,۳
	DBN	۸۱,۲	۷۸,۵

شبکه DBLSTM	MFCC	۸۳,۸	۸۲,۹
	DBN	۸۳,۵	**۸۳,۳**
HMM	MFCC	–	۷۵,۲

۶-۲- پیشنهاد برای آینده

با توجه به تجربیات و مطالعات انجام شده در این پایان‌نامه، پیشنهادهایی جهت ادامه کار وجود دارد که به‌صورت زیر می‌باشد:

۱. یکی از راه‌های پیشنهادی جهت بهبود دقت سیستم، استفاده از شبکه عصبی بازگشتی مبدل[۱][۱۳] می‌باشد که همانند CTC یک الگوریتم سربه‌سر است و مستقیما سیگنال ورودی را به دنباله واج متناظر نگاشت می‌کند. شبکه عصبی بازگشتی مبدل در واقع دو شبکه عصبی مجزا را ترکیب می‌کند که یکی از آن‌ها یک شبکه عصبی شبیه به CTC می‌باشد و شبکه رونوشت[۲] نام دارد و دیگری یک شبکه عصبی بازگشتی (که می‌تواند LSTM باشد) است و شبکه پیش‌بینی[۳] نامیده می‌شود. شبکه پیش‌بینی هر واج دنباله هدف را بر اساس واج قبلی داده شده پیش‌بینی می‌کند. با توجه به این‌که نتایج به‌دست آمده روی مجموعه دادگان TIMIT نشان‌دهنده بهبود دقت تشخیص واج در مقایسه با CTC می‌باشد [۱۳] انتظار می‌رود استفاده از این شیوه روی مجموعه فارس‌دات نیز موجب افزایش دقت گردد.

۲. روش پیشنهادی دوم استفاده از الگوریتم حذف تصادفی می‌باشد [۸۸]. این الگوریتم برای حل مشکل بیش برازش در شبکه‌های عصبی عمیق ارائه شده است و در طول فرآیند آموزش به‌صورت تصادفی تعدادی از نرون‌های لایه‌های پنهان مختلف حذف می‌شود. با توجه به این‌که استفاده از این الگوریتم در حوزه‌های مختلف از جمله بازشناسی گفتار موجب افزایش کارایی سیستم گردیده است، انتظار می‌رود استفاده از آن در شبکه DBLSTM باعث بهبود دقت مدل گردد.

۳. روش پیشنهادی دیگر ترکیب مدل مخفی مارکوف و شبکه DBLSTM می‌باشد. در این روش از شبکه DBLSTM به‌عنوان مدل آوایی استفاده می‌گردد. با توجه به این‌که استفاده از این روش ترکیبی روی مجموعه

---

[۱] Recurrent Neural Network Transducer

[۲] Transcriptipn Network

[۳] Prediction Network

TIMIT باعث افزایش دقت تشخیص واج در مقایسه با CTC شده است [۵]، انتظار می‌رود استفاده از این روش روی دادگان فارس‌دات نیز باعث به‌دست آمدن نتایج بهتری شود.

مراجع

1. Yu, D. and L. Deng, *Automatic Speech Recognition*. Signals and Communication Technology. 2015: Springer London.
2. Rabiner, L., *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, 1989. 77(2): p. 257-286.
3. Tebelskis, J., *Speech recognition using neural networks*. 1995, Siemens AG.
4. Haykin, S. and N. Network, *A comprehensive foundation.* Neural Networks, 2004. 2(2004).
5. Graves, A., N. Jaitly, and A.-r. Mohamed. *Hybrid speech recognition with deep bidirectional LSTM*. in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. 2013. IEEE.
6. Graves, A., A.-r. Mohamed, and G. Hinton. *Speech recognition with deep recurrent neural networks*. in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. 2013. IEEE.

۷. ارومیه حمیدرضا، استفاده از ویژگی‌های پی در پی برای بهبود نرخ بازشناسی گفتار و آزمون روش روی یک پایگاه داده محدود فارسی، پایان‌نامه کارشناسی ارشد، دانشگاه خواجه نصیرالدین طوسی، ۱۳۹۲.

۸. آزادی یزدی سامان، یادگیری ژرف برای بازشناسی گفتار، پایان‌نامه کارشناسی ارشد هوش مصنوعی، دانشگاه صنعتی شریف، ۱۳۹۲.

9. Gers, F., *Long short-term memory in recurrent neural networks.* Unpublished PhD dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2001.
10. Hochreiter, S. and J. Schmidhuber, *Long short-term memory*. Neural computation, 1997. 9(8): p. 1735-1780.
11. Graves, A. and J. Schmidhuber, *Framewise phoneme classification with bidirectional LSTM and other neural network architectures.* Neural Networks, 2005. 18(5): p. 602-610.
12. Graves, A., S. Fernández, and J. Schmidhuber, *Bidirectional LSTM networks for improved phoneme classification and recognition*, in *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*. 2005, Springer. p. 799-804.
13. Graves, A., *Sequence transduction with recurrent neural networks.* arXiv preprint arXiv:1211.3711, 2012.
14. Brown, P.F., et al., *Class-based n-gram models of natural language.* Computational linguistics, 1992. 18(4): p. 467-479.
15. Bijankhan, M., J. Sheikhzadegan, and M. Roohani. *FARSDAT-The speech database of Farsi spoken language*. 1994. Proccedings Australian Conference on Speech Science and Technology.

16. Huang, X., et al., *Spoken language processing: A guide to theory, algorithm, and system development*. 2001: Prentice Hall PTR.

17. Liu, Y., S. Zhou, and Q. Chen, *Discriminative deep belief networks for visual data classification.* Pattern Recognition, 2011. 44(10): p. 2287-2296.

۱۸. ویسی هادی، حجی مانی آرمیتا، بازشناسی گفتار فارسی با استفاده از شبکه عصبی حافظه کوتاه مدت ماندگار، بیست و یکمین کنفرانس ملی سالانه انجمن کامپیوتر ایران، ۱۳۹۴

۱۹. دانشور محمد، بازشناسی گفتار فارسی با استفاده از شبکه عصبی حافظه کوتاه مدت ماندگار، پایان‌نامه کارشناسی ارشد، دانشگاه تهران، ۱۳۹۵

۲۰. صامتی حسین، ویسی هادی، پژوهش نامه بازشناسی خودکار گفتار برای زبان فارسی، شورای عالی اطلاع‌رسانی، ۱۳۸۶.

21. Davis, K., R. Biddulph, and S. Balashek, *Automatic recognition of spoken digits.* The Journal of the Acoustical Society of America, 1952. 24(6): p. 637-642.

22. Olson, H.F. and H. Belar, *Phonetic typewriter.* The Journal of the Acoustical Society of America, 1956. 28(6): p. 1072-1081.

23. Fry, D., *Theoretical aspects of mechanical speech recognition.* Journal of the British Institution of Radio Engineers, 1959. 19(4): p. 211-218.

24. Sakai, T. and S. Doshita. *The Phonetic Typewriter*. in *IFIP Congress*. 1962.

25. Nagata, K., *Spoken digit recognizer for Japanese language.* NEC research & development, 1963(6)

26. Martin, T.B., A. Nelson, and H. Zadell, *Speech Recognition by Feature-Abstraction Techniques* 1964, DTIC Document.

27. Vintsyuk, T.K., *Speech discrimination by dynamic programming.* Cybernetics and Systems Analysis, 1968. 4(1): p. 52-57.

28. Reddy, D.R., *Approach to computer speech recognition by direct analysis of the speech wave.* The Journal of the Acoustical Society of America, 1966. 40(5): p. 1273-1273.

29. Itakura, F., *Minimum prediction residual principle applied to speech recognition.* IEEE Transactions on Acoustics, Speech, and Signal Processing, 1975. 23(1): p. 67-72.

30. Furui, S., *50 years of progress in speech and speaker recognition research.* ECTI Transactions on Computer and Information Technology (ECTI-CIT), 2005. 1(2): p. 64-74.

31. Rabiner, L., et al., *Speaker-independent recognition of isolated words using clustering techniques.* IEEE Transactions on Acoustics, Speech, and Signal Processing, 1979. 27(4): p. 336-349.

32. Lowerre, B. *The Harpy speech understanding system*. in *Readings in speech recognition*. 1990. Morgan Kaufmann Publishers Inc.

33. Sakoe, H., *Two-level DP-matching--A dynamic programming-based pattern matching algorithm for connected word recognition.* IEEE Transactions on Acoustics, Speech, and Signal Processing, 1979. 27(6): p. 588-595.

34. Lee, C.-H. and L.R. Rabiner, *A frame-synchronous network search algorithm for connected word recognition.* IEEE Transactions on Acoustics, Speech, and Signal Processing, 1989. 37(11): p. 1649-1658.

35. Ferguson, J., *Hidden Markov models for speech.* IDA, Princeton, NJ, 1980.

36. Lee, K.-F., H.-W. Hon, and R. Reddy, *An overview of the SPHINX speech recognition system.* IEEE Transactions on Acoustics, Speech, and Signal Processing, 1990. 38(1): p. 35-45.

37. Anusuya, M. and S.K. Katti, *Speech recognition by machine, a review.* arXiv preprint arXiv:1001.2267, 2010.

38. Trentin, E. and M. Gori, *A survey of hybrid ANN/HMM models for automatic speech recognition.* Neurocomputing, 2001. 37(1): p. 91-126.

39. Leggetter, C.J. and P.C. Woodland, *Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models.* Computer Speech & Language, 1995. 9(2): p. 171-185.

40. Gales, M. and S. Young, *Parallel model combination for speech recognition in noise*. 1993: University of Cambridge, Department of Engineering.

41. Liu, Y., et al. *Structural metadata research in the EARS program*. in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*. 2005. IEEE.

42. Hakkani-Tür, D., G. Riccardi, and A. Gorin. *Active learning for automatic speech recognition*. in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. 2002. IEEE.

43. Graves, A., et al. *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. in *Proceedings of the 23rd international conference on Machine learning*. 2006. ACM.

44. Dahl, G.E., et al. *Large vocabulary continuous speech recognition with context-dependent DBN-HMMs*. in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. 2011. IEEE.

45. Abdel-Hamid, O., et al. *Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition*. in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. 2012. IEEE.

46. Sainath, T.N., et al. *Convolutional, long short-term memory, fully connected deep neural networks*. in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. 2015. IEEE.

47. Zhang, Y., et al., *Towards end-to-end speech recognition with deep convolutional neural networks.* arXiv preprint arXiv:1701.02720, 2017.

۴۸. بی‌جن خان محمود، شیخ‌زادگان جواد، دادگان‌های گفتاری زبان فارسی، دومین کارگاه پژوهشی زبان فارسی و رایانه، ۱۳۸۵.

۴۹. الماس گنج فرشاد، سید صالحی سید علی، بی‌جن‌خان محمود، ثامتی حسین، شیخ‌زادگان جواد، شنوا-۱ سیستم بازشناسی گفتار پیوسته فارسی، نهمین کنفرانس مهندسی برق ایران، ۱۳۷۹.

۵۰. الماس گنج فرشاد، نرم افزار بازشناسی گفتار پیوسته فارسی شنوا ۲، اولین کارگاه پژوهشی زبان فارسی و رایانه، ۱۳۸۳.

۵۱. بی‌جن‌خان محمود، شیخ‌زادگان جواد، قاصدی محمد اسماعیل و همکاران، مجموعه گزارشات فنی مربوط به پروژه فارس‌دات بزرگ مستقیم، پژوهشکده پردازش هوشمند علائم، ۱۳۸۳.

52. Bijankhan, M., et al., *Lessons from building a Persian written corpus: Peykare.* Language resources and evaluation, 2011. 45(2): p. 143-164.

۵۳. بی‌جن‌خان محمود، شیخ‌زادگان جواد، هاشم الحسینی و همکاران، مجموعه گزارشات فنی مربوط به پروژه فارس-دات بزرگ مکالمه تلفنی – دیالوگ، پژوهشکده پردازش هوشمند علائم، ۱۳۸۵.

54. Sameti, H., et al., *A large vocabulary continuous speech recognition system for Persian language.* EURASIP Journal on Audio, Speech, and Music Processing, 2011. 2011(1): p. 1-12.

55. ASR Gooyesh Pardaz (AGP): http://asr-gooyesh.com/fa/ (visited on 2/12/2017)

56. Research Center of Intelligent Signal Processing (RCISP): http://www.rcisp.ac.ir/ (visited on 2/12/2017)

۵۷. حسین‌زاده هرویان مهدی، خادمیان مهدی، سید صالحی سید علی، کاربرد شبکه‌های عصبی دوسویه در تشخیص گفتار، چهاردهمین کنفرانس مهندسی پزشکی ایران، ۱۳۸۶.

۵۸. گودرزی محمد حسن، بازشناسی مقاوم به نویز گفتار بر پایه روش‌های ویژگی گمشده، پایان‌نامه کارشناسی ارشد بیوالکتریک، دانشگاه صنعتی امیر کبیر، ۱۳۸۸.

۵۹. محمد بحرانی، ارائه یک مدل زبانی ترکیبی برای بهبود عملکرد سیستم‌های بازشناسی گفتار پیوسته، پایان‌نامه دکتری هوش مصنوعی، دانشگاه صنعتی شریف، ۱۳۸۹.

۶۰. باباعلی باقر، مقاوم‌سازی سیستم‌های بازشناسی گفتار بر مبنای روش‌های جبران داده و تئوری ویژگی‌های گم شده، پایان‌نامه کارشناسی ارشد هوش مصنوعی، دانشگاه صنعتی شریف، ۱۳۸۹.

۶۱. حسینی هادی، بازشناسی گفتار فی‌البداهه-محاوره‌ای و تبدیل آن به گفتار رسمی، پایان‌نامه کارشناسی ارشد هوش مصنوعی، دانشگاه صنعتی امیر کبیر، ۱۳۸۹.

۶۲. کبودیان سید جهانشاه، بهبود مدل آکوستیکی مبتنی بر مدل پنهان مارکف، پایان‌نامه دکتری هوش مصنوعی و رباتیک، دانشگاه صنعتی امیر کبیر، ۱۳۸۹.

۶۳. قوینلی‌کر صونا، بهبود نرخ بازشناسی گفتار در شرایط نویزی با استفاده از روش‌های غیرخطی تبدیل ویژگی، پایان‌نامه کارشناسی ارشد الکترونیک، دانشگاه گیلان، ۱۳۹۱.

۶۴. محمد بافکار، مقاوم‌سازی سیستم بازشناسی گفتار پیوسته، پایان‌نامه کارشناسی ارشد هوش مصنوعی، دانشگاه تربیت معلم،۱۳۹۲.

۶۵. غدیری‌نیا مرضیه، طراحی و بهبود یک سامانه‌ی تشخیص اصطلاحات گفتاری، پایان‌نامه کارشناسی ارشد هوش مصنوعی، دانشگاه صنعتی شریف، ۱۳۹۳.

۶۶. احدی محمد، ارائه یک ساختار جدید وابسته به بافت برای بازشناسی گفتار پیوسته، پایان‌نامه کارشناسی ارشد، دانشگاه قم، ۱۳۹۳.

۶۷. شیخ الشریعه محمد، استفاده همزمان از MFCC و اطلاعات فاز جهت تشخیص گفتار زبان فارسی، پایان‌نامه کارشناسی ارشد، دانشگاه آزاد اسلامی، ۱۳۹۴.

68. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics, 1943. 5(4): p. 115-133.

69. Hebb, D.O., *The organization of behavior: A neuropsychological approach*. 1949: John Wiley & Sons.

70. Rosenblatt, F., *The perceptron: A probabilistic model for information storage and organization in the brain.* Psychological review, 1958. 65(6): p. 386.

71. Rosenblatt, F., *Two theorems of statistical separability in the perceptron*. 1958: United States Department of Commerce.

72. Rosenblatt, F., *Principles of neurodynamics.* 1962.

73. Widrow, B. and M.E. Hoff. *Adaptive switching circuits*. in *IRE WESCON convention record*. 1960. New York.

74. Kohonen, T., *Correlation matrix memories.* IEEE transactions on computers, 1972. 100(4): p. 353-359.

75. Anderson, J.A., et al., *Distinctive features, categorical perception, and probability learning: Some applications of a neural model.* Psychological review, 1977. 84(5): p. 413.

76. Kohonen, T., *Self-organized formation of topologically correct feature maps.* Biological cybernetics, 1982. 43(1): p. 59-69.

77. Hinton, G.E. and T.J. Sejnowski. *Optimal perceptual inference*. in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1983. Citeseer.

78. Parker, D.B., *Learning logic.* 1985.

79. Le Cun, Y., *Learning process in an asymmetric threshold network*, in *Disordered systems and biological organization*. 1986, Springer. p. 233-240.

80. Hopfield, J.J., *Neural networks and physical systems with emergent collective computational abilities.* Proceedings of the national academy of sciences, 1982. 79(8): p. 2554-2558.

81. Carpenter, G.A. and S. Grossberg, *ART 2: Self-organization of stable category recognition codes for analog input patterns.* Applied optics, 1987. 26(23): p. 4919-4930.

82. Fukushima, K., S. Miyake, and T. Ito, *Neocognitron: A neural network model for a mechanism of visual pattern recognition.* IEEE Transactions on Systems, Man, and Cybernetics, 1983(5): p. 826-834.

83. Fukushima, K., *Cognitron: A self-organizing multilayered neural network.* Biological cybernetics, 1975. 20(3-4): p. 121-136.

84. Elman, J.L., *Finding structure in time.* Cognitive science, 1990. 14(2): p. 179-211.

85. Jordan, M.I., *Attractor dynamics and parallellism in a connectionist sequential machine.* 1986.

86. Schuster, M. and K.K. Paliwal, *Bidirectional recurrent neural networks.* Signal Processing, IEEE Transactions on, 1997. 45(11): p. 2673-2681.

87. Snoek, J., H. Larochelle, and R.P. Adams. *Practical bayesian optimization of machine learning algorithms*. in *Advances in neural information processing systems*. 2012.

88. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting.* Journal of Machine Learning Research, 2014. 15(1): p. 1929-1958.

89. Deng, L. and J. Chen. *Sequence classification using the high-level features extracted from deep neural networks*. in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. 2014. IEEE.

90. Fausett, L., *Fundamentals of neural networks: architectures, algorithms, and applications*. 1994: Prentice-Hall, Inc.

91. Lippmann, R., *An introduction to computing with neural nets.* IEEE Assp magazine, 1987. 4(2): p. 4-22.

92. Sandberg, I.W., *Nonlinear dynamical systems: feedforward neural network perspectives*. Vol. 21. 2001: John Wiley & Sons.

93. Engelbrecht, A.P., *Computational intelligence: an introduction*. 2007: John Wiley & Sons.

94. Graves, A., *Supervised sequence labelling with recurrent neural networks*. Vol. 385. 2012: Springer.

95. Cruse, H., *Neural networks as cybernetic systems*. 1996: Thieme Stuttgart.

96. Schuster, M., *On supervised learning from sequential data with applications for speech recognition.* Daktaro disertacija, Nara Institute of Science and Technology, 1999.

97. Hinton, G.E. and R.R. Salakhutdinov, *Reducing the dimensionality of data with neural networks.* Science, 2006. 313(5786): p. 504-507.

98. Fischer, A. and C. Igel. *An introduction to restricted Boltzmann machines*. in *Iberoamerican Congress on Pattern Recognition*. 2012. Springer.

99. Lee, H., C. Ekanadham, and A.Y. Ng. *Sparse deep belief net model for visual area V2*. in *Advances in neural information processing systems*. 2008.

100. Carreira-Perpinan, M.A. and G. Hinton. *On Contrastive Divergence Learning*. in *AISTATS*. 2005. Citeseer.

101. Salakhutdinov, R., *Learning deep generative models*. 2009, University of Toronto.

102. Young, S.J. and S. Young, *The HTK hidden Markov model toolkit: Design and philosophy*. 1993: University of Cambridge, Department of Engineering.

Abstract

The process of converting speech signal to its equivalent text is known as Automatic Speech Recognition (ASR). The most important methods for speech recognition are Hidden Markov Model (HMM) and Artificial Neural Network (ANN). One way to increase the accuracy of a speech recognition system is improving the quality of Acoustic Modeling (AM). In this thesis, for the first time, we have used deep unidirectional and bidirectional Long Short Term Memory (LSTM) neural network with Connectionist Temporal Classification (CTC) output layer to create Persian acoustic models. Because of the sequential structure of speech signal, recurrent neural networks are appropriate for processing them. However, because of vanishing problem of recurrent neural networks they are not suitable for processing long sequential data. LSTM as a recurrent neural network, has solved the vanishing problem by replacing hidden layer neurons with memory blocks.

Moreover, in this thesis we have used Deep Belief Network (DBN) for feature extraction and compared the results with the baseline feature extraction method, Mel Frequency Cepstral Coefficient (MFCC).

The results show that, the accuracy of phoneme recognition is improved by using DBN features in comparision with the MFCC. Also, deep bidirectional LSTM with DBN features has improved the Persian phoneme recogntion rate about 8.1% in comparison with the HMM on Farsdat speech dataset.

Keywords

Persian speech recognition, Long short term memory neural network, Bidirectional neural network, Deep neural network, Recurrent neural network, Connectionist temporal classification

University of Tehran
Faculty of New Sciences and Technologies
Interdisciplinary Technology Group (Network Sciences and Technologies)

Persian Speech Recognition using Deep Learning

By:
Armita Hajimani

Supervisor:
Dr. Hadi Veisi

A thesis submitted to the Graduate Office in Fulfillment of Requirements for the Degree of Master of Science in Decision Science and Knowledge Engineering

March 2017