عنوان پروژه (کارشناسی ارشد)

تشخیص احساسات گفتاری با استفاده از شبکه عصبی کانولوشن

پروژه/ پایان‌نامه/ رساله برای دریافت درجه کارشناسی/ ارشد/ دکتری

در رشته مهندسی / گرایش

نام دانشجو:

استاد (اساتید) راهنما:

جنابت آقای

استاد (اساتید) مشاور:

مهر ماه ۱۴۰۰

تصویر صورتجلسه دفاع از پروژه/ پایان‌نامه/ رساله

تأییدیه هیئت داوران جلسه دفاع از پروژه/ پایان‌نامه/ رساله

دانشکده مهندسی

نام دانشجو:

عنوان پروژه/ پایان‌نامه/ رساله:

تاریخ دفاع:

رشته مهندسی گرایش:

ردیف	سمت	نام و نام خانوادگی	مرتبه دانشگاهی	دانشگاه یا مؤسسه	امضا
۱	استاد راهنمای اول				
۲	استاد راهنمای دوم				
۳	استاد مشاور اول				
۴	استاد مشاور دوم				
۵	استاد مدعو خارجی اول				
۶	استاد مدعو خارجی دوم				
۷	استاد مدعو داخلی اول				
۸	استاد مدعو داخلی دوم				

مجوز بهره‌برداری از پروژه/ پایان‌نامه/ رساله

بهره‌برداری از این پروژه/ پایان‌نامه/ رساله در چهارچوب مقررات کتابخانه و باتوجه به محدودیتی که توسط استاد راهنما به‌شرح زیر تعیین می‌شود، بلامانع است:

☐ بهره‌برداری از این پروژه/ پایان‌نامه/ رساله برای همگان بلامانع است.

☐ بهره‌برداری از این پروژه/ پایان‌نامه/ رساله با اخذ مجوز از استاد راهنما، بلامانع است.

☐ بهره‌برداری از این پروژه/ پایان‌نامه/ رساله تا تاریخ ..... ممنوع است.

نام استاد (اساتید) راهنما:

تاریخ:

امضا:

تقدیم به:

تشکر و قدردانی:

با تشکر از زحمات استاد گرانقدر جناب

چکیده

موضوع پایان نامه پیش‌بینی و تشخیص احساسات گفتاری با استفاده از شبکه عصبی عمیق کانولوشنال است . تشخیص احساسات گفتاری (SER) یکی از چالش برانگیزترین کارها در حوزه تجزیه و تحلیل سیگنال گفتار است ، این یک چالش حوزه تحقیقاتی است که سعی می کند احساسات را از سیگنال های گفتاری استنباط کند. دقت در تشخیص احساساتی اعم از شادی، غم، ترس، عصبانیت و آرامش بین زنان و مردان بویژه در زمانی که سیگنالهای صدا دارای سرو صدا و نویز پس زمینه است و یا احساسات در گفتار بدرستی و با شفافیت منعکس نمی شود، کار پیچیده ای می باشد. یک شبکه عصبی پیچیده (کانولوشنال) تحت نظارت عمیق برای طبقه بندی هر احساس در صداهای جمع آوری شده آموزش دیده است. با بهره گیری از کتابخانه تنسورفلو Tensorflow و کراسKeras لایه های متعدد در محیط Google Colab برای مدل شکل گرفته است. مجموعه با تعداد ۲۵ Actor متشکل از صداهای زنان و مردان با احساسات مختلف جمع آوری شده از بانک Ravdess مورد آزمایش قرار می گیرد. در آخر بخش مهم اجرای مدل شبکه عصبی پیچیده (کانولوشنال) و تست مدل است که با اندازه گیری متریک دقت میتوان به میزان دقت در مدل واقف شد.

واژه‌های کلیدی: شبکه عصبی عمیق کانولوشنال ، تشخیص احساسات، Tensorflow، Keras، احساسات گفتاری.

فهرست مطالب

Contents

فهرست علائم اختصاری

شبکه عصبی کانولوشنال Convolutional Neural Network ..... *CNN*

فصل ۱:

مقدمه

۱-۱- معرفی

تشخیص احساسات مبتنی بر گفتار دارای مزایای کاربردی عملی فراوانی است. در واقع تشخیص احساسات فرایند شناسایی احساسات انسانی است. دقت افراد در تشخیص احساسات دیگران بسیار متفاوت است. استفاده از این فناوری برای کمک به افراد در تشخیص احساسات یک حوزه تحقیقاتی نسبتاً نوپا است. به طور کلی ، این فناوری در صورتی که از چند روش در زمینه استفاده کند ، بهترین عملکرد را دارد. تا به امروز ، بیشترین کار بر روی تشخیص خودکار حالات چهره از طریق ویدئو ، عبارات گفتاری از طریق صدا ، عبارات نوشتاری از متن اندازه گیری می شود.

تشخیص احساسات یکی از مهمترین استراتژی های بازاریابی در دنیای امروز است. شما می توانید موارد مختلف را برای یک فرد به طور خاص متناسب با علاقه خود شخصی سازی کنید.به همین دلیل ، بنای این تحقیقاین است که بتوان احساسات افراد را فقط با صدای آنها تشخیص داد که اجازه می دهد بسیاری از برنامه های مرتبط با هوش مصنوعی مدیریت شود. برخی از مثالها می تواند شامل مراکز تماس برای پخش موسیقی هنگام عصبانی شدن فرد در تماس باشد. یکی دیگر می تواند یک ماشین هوشمند باشد که هنگام عصبانیت یا ترس سرعت خود را کاهش می دهد.در نتیجه این نوع برنامه دارای پتانسیل زیادی در جهان است که می تواند به نفع شرکت ها و حتی ایمنی مصرف کنندگان باشد.

رابط کاربری صوتی (VUI) تعامل گفتاری انسان با رایانه ها را ممکن می سازد ، از تشخیص گفتار برای درک دستورات گفتاری و پاسخ به سوالات استفاده می کند و معمولاً برای پخش پاسخ از متن به گفتار استفاده می کند. دستگاه فرمان صوتی (VCD) دستگاهی است که با رابط کاربری صوتی کنترل می شود.

SER اگرچه چندان محبوب نیست ، SER در این سالها حوزه های زیادی را وارد این عرصه کرده است ، از جمله:

حوزه پزشکی: در دنیای پزشکی از راه دور که بیماران در بسترهای تلفن همراه مورد ارزیابی قرار می گیرند ، توانایی یک متخصص پزشکی در تشخیص احساس بیمار در واقع می تواند در روند بهبود مفید باشد.

خدمات به مشتریان: در مرکز تماس از مکالمه برای تجزیه و تحلیل مطالعه رفتاری همراهان تماس با مشتریان استفاده می شود که به بهبود کیفیت خدمات کمک می کند.

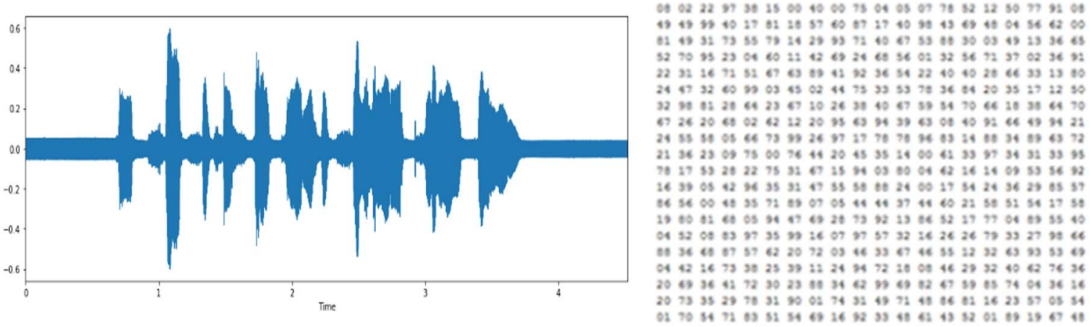
سیستم های توصیه گر: توصیه می شود محصولات را بر اساس احساسات مشتریان نسبت به آن محصول به مشتریان توصیه شود.

یکی از مولفه های مهم این فرآیند تشخیص گفتار با استفاده از شبکه عصبی کانولوشنال است ، که کار زیادی را طلب می کند و نیاز به تخصص در این حوزه دارد. به تازگی ، تکنیک های تشخیص گفتاری رایانه ای و یادگیری ماشینی با موفقیت برای بررسی خودکار احساسات انسانی بر روی صداهای ضبط شده استفاده شده است. تعداد زیادی از مقالات و تحقیقات اخیر در تشخیص احساسات و توصیف حالت های انسانی به وضوح نشان دهنده علاقه روزافزون به این حیطه تحقیقاتی است.

۲-۱- شبکه عصبی کانولوشن

شبکه عصبی پیچشی یا شبکه عصبی کانولوشن، از مهم‌ترین نوآوری‌ها در حوزه‌ی بینایی کامپیوتر به حساب می‌آیند. لغت شبکه عصبی در سال ۲۰۱۲، معروفیت فراوانی کسب کرد؛ در این سال الکس چریشفسکی، با استفاده از شبکه عصبی توانست برنده جایزه ImageNet (المپیک سالیانه بینایی کامپیوتر) شود. ریشفسکی توانست خطای دسته‌بندی (classification) را از ۲۶ درصد به ۱۵ درصد کاهش دهد. این کاهش در آن زمان بسیار چشم‌گیر بود. از آن زمان، شرکت‌های متعددی از یادگیری عمیق به عنوان هسته اصلی محصولات خود استفاده کرده‌اند. گوگل، فیسبوک، آمازون، اینستاگرام و پینترست از شبکه عصبی استفاده می‌کند تا تصاویر را به صورت خودکار تگ‌گذاری نماید؛ با این حال بیش‌ترین استفاده‌ی شبکه عصبی در پردازش تصویر است. در این بخش از معرفی متد تحقیق، به چیستی شبکه عصبی کانولوشن پرداخته خواهد شد و اینکه چگونه از آن در دسته‌بندی تصاویر استفاده می‌شود.

دسته‌بندی تصاویر یا اصوات در واقع پروسه ای است که در آن تعدادی تصویر یا صوت را از ورودی می‌گیریم و در خروجی، کلاس آن‌ها (نوع احساسات صوت، مثلا در اینجا کلاس به احساساتی از قبیل شادی، غم، ترس، عصبانیت و آرامش بین زنان و مردان اطلاق می شود) یا درصد احتمال تعلق به هر کلاس را مشخص می‌کنیم. انجام چنین عملی، یعنی تشخیص و نام‌گذاری (labeling) اصوات، کلاسیفیکیشن نامیده می شود.



شکل (۱-۱) تصویر سمت چپ تصویر واقعی است که چشمان ما در هنگام پخش فایل صوتی در پلیرها می بیند و تصویر سمت راست چیزی است که کامپیوتر برای آنالیز می بیند.

فصل ۲:

پیشینهٔ تحقیق

مقدمه

۱-۲- یادگیری عمیق

در یک تعریف کلی، یادگیری عمیق، همان یادگیری ماشین است، به طوری که در سطوح مختلف نمایش یا انتزاع[1] یادگیری را برای ماشین انجام میدهد. با این کار، ماشین درک بهتری از واقعیت وجودی داده ها پیدا کرده و میتواند الگوهای مختلف را شناسایی کند. مدل های یادگیری عمیق به شکلی نه چندان روشن از الگوهای پردازش اطلاعاتی و ارتباطی در سیستم های عصبی زیستی الهام گرفته شده اند اما تفاوت های مختلفی در ویژگی های ساختاری و عملکردی با مغزهای زیستی (به ویژه مغز انسان) دارند، که باعث عدم همخوانی آنها با شواهد علوم اعصاب میشود. یادگیری عمیق[2] به دنبال هوش مصنوعی پا به عرصه حضور گذاشته است. این یادگیری به یاری هوش مصنوعی آمده است تا به شکلی طبیعی تر به نیاز ها و خواست های بشر واکنش نشان دهد. هوش مصنوعی در جهت یاری رساندن به بشر امروزی روی کار آمده است. سالیان درازی از روی کار آمدن هوش مصنوعی نمی گذرد. اما در طی همین زمان کوتاه بشر در زمینه های متفاوتی از این تکنولوژی بهره برده است. یادگیری عمیق، دسته ای از الگوریتم های یادگیری ماشین است که:

از آبشاری از لایه های چندگانهِ واحدهای پردازش غیرخطی برای استخراج و تبدیل ویژگی استفاده میکنند. هر لایه تالی، از خروجی لایه قبل به عنوان ورودی استفاده میکند.
به شکلی نظارت شده (مثل طبقه بندی) و یا بدون نظارت (مثل تحلیل الگو) یادگیری میکنند.
لایه های چندگانه ای از نمایش را یادگیری میکنند که متناظر با سطوح مختلفی از انتزاعات هستند؛ این سطوح سلسله ای از مفاهیم را تشکیل میدهند.

یادگیری عمیق یک نوع شبکه عصبی بوده که فراداده[3] را به عنوان یک ورودی جذب می‌کند و داده‌ها ورودی را از طریق برخی لایه‌های تبدیل غیرخطی پردازش و محاسبه کرده و به عنوان داده‌های خروجی برمی‌گرداند. این الگوریتم دارای یک ویژگی منحصربفرد بوده که آن ویژگی استخراج خودکار[4] محسوب می‌شود. این بدین معنی است که الگوریتم ویژگی‌های موردنیاز و مرتبط را جهت حل مشکل درک می‌کند. این موجب کاهش وظیفه برنامه نویسان شده تا به انتخاب صریح ویژگی‌ها بپردازند. این الگوریتم حتی برای

[1] Abstraction
[2] Deep Learning
[3] Metadata
[4] Feature Extraction

حل چالش ها تحت نظارت، بدون نظارت به کار گرفته می شود. در یادگیری عمیق هر لایه پنهان مسئول آموزش مجموعه‌ای از ویژگی‌های منحصربفرد بوده که براساس خروجی لایه پیشین عمل می کند. با افزوده شدن بر تعداد لایه های پنهان، پیچیدگی داده‌ها بیشتر شده و مشکلات را افزایش می‌دهد. همچنین این نوع یادگیری سلسله مراتبی، ویژگی‌های سطح پایین را به ویژگی‌های سطح بالا تبدیل می‌کند. با چنین کاری الگوریتم یادگیری عمیق مورد استفاده قرار گرفته و به حل مشکلات پیچیده که لایه های غیرخطی متعددی را دربرمی گیرد، می پردازد. در یادگیری عمیق، هر سطح یاد میگیرد که داده های ورودی خود را به یک نمایش اندکی مجردتر و ترکیبی تر تبدیل کند. در یک کاربرد شناسایی تصویر، ورودی خام میتواند ماتریسی از پیکسل ها باشد؛ اولین لایه نمایشی ممکن است پیکسل ها را مجرد کند و لبه ها را کدگذاری کند؛ لایه دوم ممکن است چینش لبه ها را بسازد و کدگذاری کند؛ لایه سوم ممکن است بینی و چشم ها را کدگذاری کند؛ و لایه چهارم ممکن است تشخیص دهد که تصویر، شامل یک چهره است. چیزی که اهمیت دارد، این است که یک پروسه یادگیری عمیق، به خودی خود میتواند یاد بگیرد که کدام ویژگی ها بطور بهینه در کدام سطح قرار دهد. در سال های اخیر، یادگیری عمیق، تحول بزرگی را در یادگیری ماشین و هوش مصنوعی ایجاد کرده است. از سال ۲۰۱۲ تا کنون، تمامی رتبه های برتر چالش شناسایی بصری ImageNet، که به جام جهانی بینایی ماشین معروف است، از شبکه های عصبی عمیق استفاده کرده اند. همچنین، تمام روش های برتر در رقابت های دسته بندی تصاویر اعداد دست نویس MNIST (با ۲۱ خطا در ۱۰,۰۰۰ تصویر) و تصاویر طبیعی CIFAR (با خطای کمتر از ۵٪) نیز به مدل های شبکه عصبی عمیق تعلق دارد. از سال ۲۰۱۲ به بعد، شرکت های بزرگ نرم افزاری و سخت افزاری مانند Google, Microsoft, NVIDIA نیز بخش مهمی از فعالیت های پژوهشی و تجاری خود را به یادگیری عمیق اختصاص داده اند. به دلیل وجود لایه های متفاوت و سطح های متفاوتی از اطلاعات از واژه عمیق استفاده می شود. شبکه عصبی از دو یا نهایتا سه لایه تشکیل شده است. در حالی که شبکه عصبی عمیق از بیش از ۱۵۰ لایه تشکیل شده است. یادگیری عمیق به گونه ای طراحی شده است که قادر باشد بدون دستور العمل هایی که توسط اپراتور صادر می شود، اطلاعات مورد نیاز خود را از میان حجم وسیعی از اطلاعات استخراج کرده و مورد استفاده قرار دهد.

۲-۲- متد شبکه عصبی کانولوشنCNN [۱]

۲-۲-۱- ورودی تصویری و خروجی کلاسه شده در یک شبکه عصبی کانولوشن

وقتی یک کامپیوتر صوت و یا تصویری را به عنوان ورودی دریافت می‌کند، آن را به صورت آرایه‌ای از اعداد می‌بیند. تعداد آرایه‌ها به سیگنال صوتی (بر اساس داده های متوالی و سری زمانی) و یا سایز تصویر (بر

اساس پیکسل) بستگی دارد. برای مثال اگر یک صوت از پیش ضبط شده فرمت wav را به کامپیوتر دهیم، آرایه جانشین آن دارای m*n خانه خواهد بود. هر کدام از خانه‌ها یا المنت ها نیز عددی بین 1- تا 1 را می‌گیرند. این عدد شدت آوا را نشان می‌دهد. این اعداد هر چند در وهله اول بی‌معنی به نظر می‌رسند، اما در پردازش صدا با استفاده از الگوریتم ها، ابزار مناسب، همین اعداد هستند. ایده اصلی آن است که به کامپیوتر یا مدل پردازش صدا، آرایه‌ای از اعداد، شبیه آن چه توضیح داده شد، داده و کامپیوتر نیز در خروجی چنین چیزی را مشخص می‌کند: این صدا با احتمال ۸۰ درصد دارای احساسات خوشحالی است، و یا با احتمال ۱۵ درصد دارای احساسات ترس است.

۲-۲-۲- شیوه عملکرد شبکه عصبی کانولوشن چیست؟

متد حل مسئله در شبکه عصبی کانولوشن دریافت صدا و یافتن ویژگی‌های منحصر به فرد آوا با استفاده از کتابخانه librosa مانند قدرت ، صدای پیکربندی و مجرای صوتی از سیگنال گفتار است. سپس تشخیص دهد در صدا، احساسی موجود است یا نه. مثلا برای تشخیصش احساسات، ابتدا به مولفه های جزئی‌تر آن مانند قدرت، طول موج توجه می کند و ضمن تطبیق با الگوهای موجود در داده های آموزشیTrain Set ، در می یابد که در چه احساسی در صدا وجود دارد. در نتیجه مدل برای درک و تشخیص احساسات در صداهای پیچیده‌ای مثل صدا همراه با نویز پس زمینه ، ابتدا ویژگی‌های (feature) ساده‌تر آن صدا مانند قدرت و تغییر طول موج را تشخیص می‌دهد. در یک شبکه عصبی، لایه‌های متعددی وجود دارند؛ در هر یک از این لایه‌ها، ویژگی‌های خاصی تشخیص داده می‌شوند و در نهایت، در لایه‌ی آخر، صدا به طور کامل شناسایی می‌شود. روندی که توضیح داده شد، فرایند کلی نحوه کار یک شبکه عصبی کانولوشن بود؛ حال به جزئیات بیش‌تری پرداخته می شود.

۳-۲-۲- ارتباط شبکه عصبی پیچشی با بیولوژی

در این قسمت مفاهیم پایه‌ای‌تر مورد بررسی قرار می گیرد. عبارت شبکه عصبی کانولوشن قرابت زیادی با زیست شناسی و نوروساینس دارد. ساختار شبکه عصبی پیچشی (CNN) در حقیقت از قشر بینایی مغز الهام گرفته شده است. در سال ۱۹۶۲، دو دانشمند با نام‌های هابل و ویزل، آزمایش جالبی انجام دادند. آن‌ها نشان دادند که با دیدن لبه‌ها با اشکال مختلف، سلول‌های خاصی در قشر بینایی مغز تحریک می‌شوند. برای

مثال با دیدن خطوط افقی، سلول‌های خاصی تحریک می‌شوند و با دیدن خطوط عمود بر هم سلول‌های متفاوتی حساسیت نشان می‌دهند. هابل و ویزل دریافتند که این سلول‌ها به شکل ستونی و خیلی منظم در کنار همدیگر قرار گرفته‌اند و حاصل همکاری آن‌ها با هم این است که انسانها می‌توانند ادراک تصویری خوبی از محیط پیرامون داشته باشند. اساس کار شبکه عصبی کانولوشن نیز مانند قشر بینایی مغز است. در حقیقت در یک CNN، لایه‌های مختلفی وجود دارند که هر یک لایه مخصوص شناسایی موارد خاصی است. در نهایت نیز خروجی مدل ادراک تصویری کامل است.

۴-۲-۲- ساختار شبکه عصبی پیچشی

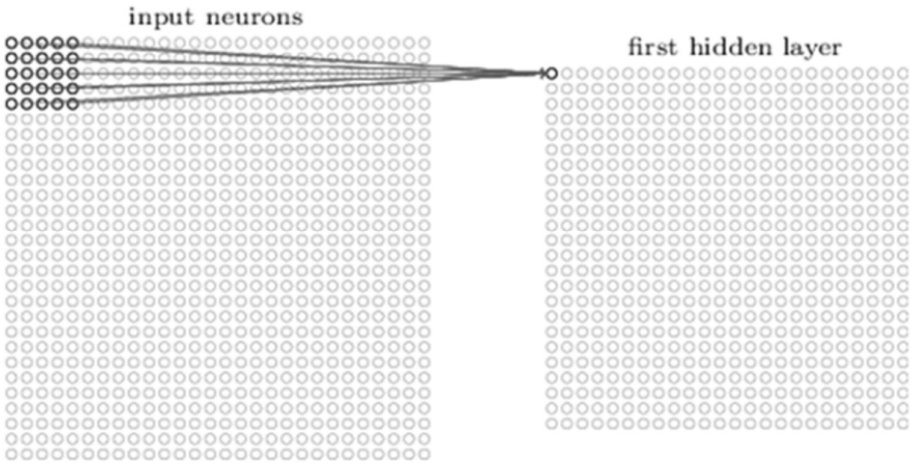
همان‌طور که اشاره شد، در یک شبکه عصبی پیچشی، کامپیوتر یک تصویر را به عنوان ورودی می‌گیرد؛ سپس این تصویر وارد یک شبکه‌ی پیچیده با چندین لایه‌ی پیچشی و غیر خطی می‌شود. در هر یک از این لایه‌ها، عملیات‌هایی انجام می‌شود و در انتها بر روی خروجی، یک کلاس یا درصد وقوع چند کلاس مختلف نشان داده می‌شود. قسمت سخت ماجرا، لایه‌های میانی و نحوه عملکرد آن‌هاست. در ادامه به بررسی مهم‌ترین لایه‌ها پرداخته می شود.

۵-۲-۲- لایه اول در شبکه عصبی کانولوشن

لایه اول در یک شبکه عصبی پیچشی، همیشه یک لایه‌ی کانولوشنال است. همان‌طور که قبلا اشاره کردیم، ورودی این لایه یک آرایه از اعداد است. لایه اول در شبکه عصبی مانند یک چراغ قوه کار می‌کند. در یک اتاق تاریک، چراغ قوه‌ای را تصور کنید که بر گوشه‌ی بالا و سمت چپ تصویر انداخته می شود و محدوده‌ای از تصویر روشن می نمایاند و آن قسمت دیده می شود. سپس چراغ قوه بر روی قسمت‌های دیگر تصویر تابانده می شود تا کم‌کم کل تصویر را روشن نمایاند. همین روند دریادگیری ماشین، رخ می دهد (گرچه این روند در تشخیص احساسات در صدا نیز رخ می دهد و هر پارت از صدا آرام آرم با استفاده از فیلترینگ، پرداطش می شود).

در شبکه عصبی کانولوشن، به این چراغ قوه، فیلتر (filter) (یا نورون یا کرنل) می‌گوییم. آن قسمتی از تصویر یا صوت که چراغ قوه به آن نور می‌تاباند، محدود پذیرش (receptive field) نام دارد. لازم به ذکر است، فیلترها نیز خود آرایه‌هایی از اعداد هستند. به اعداد موجود در فیلتر، وزن (weight) یا پارامتر

(parameter) گفته می‌شود. لازم به ذکر است که عمق این فیلتر باید با عمق صوت و یا تصویر برابر باشد. فیلتر در هر نگاه، یک قسمت از صوت و یا تصویر را می‌بیند. سپس بر روی صوت و یا تصویر حرکت می‌کند تا قسمت‌های دیگر را هم اسکن کند. به این حرکت فیلتر بر روی تصویر، پیچیدن (convolve) گفته می شود. همین طور که فیلتر از تصویر عبور می‌کند، اعداد موجود در فیلتر با آرایه عددی پیکسل‌های صوت و یا تصویر ضرب می‌شود. در نهایت نیز تمام حاصل ضرب‌ها با یکدیگر جمع می‌شوند و به یک عدد می‌رسیم.

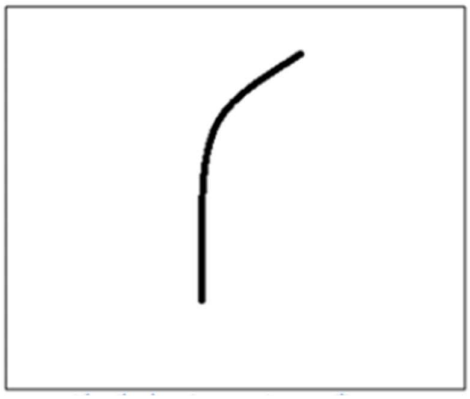


شکل (۱-۲) صوت و یا تصویر فیلتر چرخشی ۵*۵ در محدوده داده ورودی و تولید یک نقشه فعال ساز

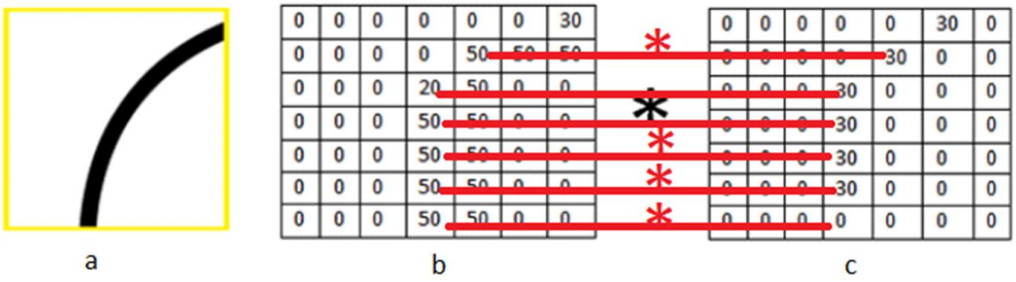
۶-۲-۲- لایه اول در شبکه عصبی کانولوشن | کاربردی

هر یک از فیلترهایی را که در قسمت قبلی به آن‌ها اشاره شد، می‌تواند به عنوان یک شناساگر ویژگی (feature identifier) در نظر گرفته شود. منظور از ویژگی (feature) در این جا، چیزهایی مانند قدرت صدا، طول موج است. فرض بر این است که فیلتر اول، یک فیلتر با ابعاد m*n و یک شناساگر قدرت صدا است. این فیلتر در حقیقت یک ماتریس عددی مانند صوت و یا تصویر زیر است که درایه‌های این ماتریس در محل‌هایی که قدرت صدا در آن وجود دارد، مقادیر عددی بالاتری دارند. حال این فیلتر را بر روی قسمتی از صوت و یا تصویر مد نظرمان قرار می‌دهیم. پس از آن مانند شکل زیر، درایه به درایه اعداد موجود در خانه‌ها را با هم ضرب و حاصل ضرب‌ها را با یکدیگر جمع خواهد شد.

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0



شکل (۲-۲) تصویر سمت چپ مقادیر درایه های قدرت صوت سمت راست را در درایه های یک ماتریس به نمایش در آورده و در قدرت صوت ، شاهد افزایش از ۰ به ۳۰ می باشیم.

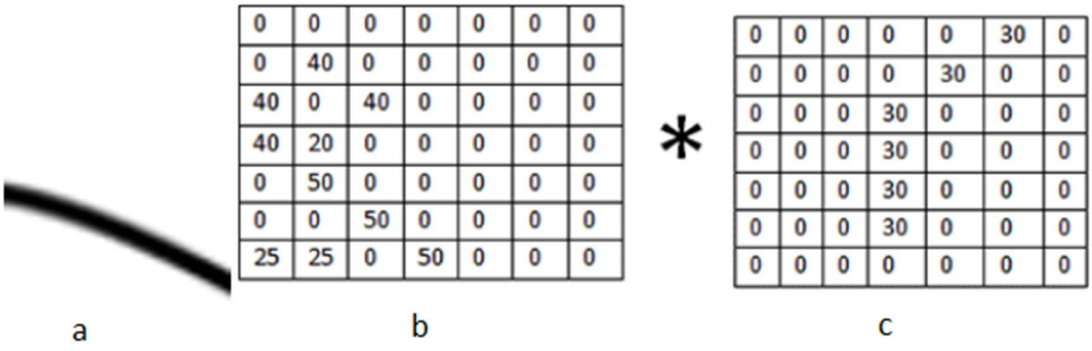


شکل (۳-۲) تصویر a نشاندهنده قدرت در صدا می باشد، تصویر b محدود پذیرش (receptive field) صدای اصلی، و تصویر c فیلتر یا نورون یا کرنل

$$Filter = (50 * 0) + (50 * 30) + (50 * 30) + (50 * 30) + (20 * 30) + (50 * 30) = 6600$$

همانطور که مشاهده می‌شود، حاصل به‌دست‌آمده، یک عدد بزرگ است. بزرگ بودن این عدد نشانگر آن است که در این ناحیه یک قدرت صدا مانند این فیلتر وجود دارد.

در تصویر زیر، حاصل ضرب عدد کوچکی می‌شود؛ علت آن است که فیلتر با صدای ورودی تطابق ندارد. هدف یافتن یک نقشه فعال‌سازی است؛ قسمت بالا و سمت چپ این نقشه فعال‌سازی، مقدار ۶۶۰۰ را خواهد داشت. این عدد بزرگ نشان‌دهنده‌ی آن است که در ناحیه‌ی خاصی از صدا، با احتمال زیاد یک قدرت وجود دارد. در این جا تنها از یک فیلتر استفاده شده است. برای آن که اطلاعات بیش‌تری از صدا استخراج شود، نیاز است تا از فیلترهای بیش‌تری استفاده شود؛ استفاده از فیلترهای بیش‌تر یعنی ابعاد بالاتر.



شکل (۴-۲) تصویر a نشاندهنده قدرت در صدا می باشد، تصویر b محدود پذیرش (receptive field) در صدای اصلی، و تصویر c فیلتر یا نورون یا کرنل

$Filter\ Result = 0$

۷-۲-۲- لایه‌های عمیق‌تر شبکه عصبی کانولوشن

در یک شبکه عصبی، علاوه بر لایه‌ی توضیح داده شده، لایه‌های دیگری نیز وجود دارند. این لایه‌ها وظایف و عملکردهای گوناگونی دارند. به طور کلی، لایه‌های داخلی، مسئول نگهداری و حفظ ابعاد و امور غیرخطی هستند. آخرین لایه در شبکه عصبی کانولوشن نیز از اهمیت خاصی برخوردار است.

۸-۲-۲- لایه آخر در شبکه عصبی پیچشی

در لایه آخر یک شبکه عصبی کانولوشن، خروجی سایر لایه‌ها، به عنوان ورودی دریافت می‌شود. خروجی لایه آخر هم یک بردار N بعدی است. N تعداد کلاس‌های موجود است. به عنوان مثال اگر شبکه ایی مانند آنچه در تشخیص احساسات گفتاری ها بر روی صدا ها داریم، احساساتی اعم از شادی، غم، ترس، عصبانیت و آرامش، در نتیجه تعداد کلاس‌ها پنج تاست؛ چون ۵ نوع کلاس، شادی، غم، ترس، عصبانیت و آرامش وجود دارد. در بردار N بعدی، هر مولفه، احتمال وقوع یک کلاس را نشان می‌دهد. کاری که لایه‌ی آخر یک شبکه عصبی کانولوشن می‌کند آن است که به ویژگی‌های لایه‌های سطح بالا نگاه می‌کند و میزان مطابقت این ویژگی‌ها را با هر کلاس مقایسه می‌کند؛ هر چه این مطابقت بیش‌تر باشد، احتمال وقوع آن کلاس، بالاتر معرفی می‌شود.

۹-۲-۲- نحوه عملکرد شبکه عصبی کانولوشن چیست؟

مدل کانولوشنال طی یک فرایند آموزش (training) می‌تواند مقادیر مناسب را به فیلترها تخصیص دهد.این فرایند backpropagation نام دارد. در ابتدای کار، اعداد موجود در ماتریس فیلتر، رندم و تصادفی هستند. به مرور زمان و با آموزش صداهای مختلف به مدل، اعداد موجود در فیلتر تصحیح می‌شوند تا به یک عملکرد قابل قبول برسند.

۱۰-۲-۲- تست شبکه عصبی CNN

پس از آن که مدل نهایی و آمده شد، وقت تست کردن فرا می‌رسد. برای تست مدل از تعدادی صدا که محتویات احساسات آن مشخص است، استفاده می‌شود. صدا را به ورودی مدل می‌دهیم تا خروجی را به ما نشان دهد؛ سپس خروجی را بررسی می‌کنیم تا ببینیم درست عمل شده است یا نه.

۱۱-۲-۲- لایه های کانولوشنال

در معماری شبکه عصبی پیچشی سنتی ، لایه های دیگری نیز وجود دارند که بین این لایه های متقاطع پراکنده شده اند. در یک مفهوم کلی ، لایه ها، توابع غیر خطی و کنترل کننده ابعاد هستند که به بهبود یکپارچگی مدل و کنترل برازش بیش از حد overfittingکمک می کند. برازش بیش از حد یا overfitting زمانی اتفاق می افتد که مدل بتواند بر اساس داده های موجود در مجموعه آموزشی طبقه بندی یا پیش بینی کند ، اما در طبقه بندی داده هایی که بر روی آنها آموزش ندیده است ، خوب عمل نمی کند. بنابراین اساساً ، مدل از داده های موجود در آموزش بیش از حد برخوردار است .معماری کلاسیک CNN شبیه این خواهد بود.

```
Input -> Conv -> ReLU -> Conv -> ReLU -> Pool -> ReLU -> Conv -> ReLU -> Pool ->Fully Connected
```

فیلترهایی که در لایه کانولوشن اول برای تشخیص حاشیه و خطوط مرزی طراحی شده اند، مورد بررسی قرار گرفت. آنها ویژگی های سطح پایین مانند قدرت صدا و طول موج ها را تشخیص می دهند. همانطور که تصور می شود ، برای پیش بینی اینکه یک صدا چه احساسی دارد، ما به شبکه نیاز داریم تا بتوانیم ویژگی های سطح بالاتری مانند اجزا و مولفه های اصلی صدا مانند نوع احساسات را تشخیص دهیم. خروجی شبکه

بعد از اولین لایه conv با حجم m*n خواهد بود (با فرض اینکه از فیلتر m * n استفاده شود). هنگامی که از یک لایه دیگر متقاطع عبور شود، خروجی اولین لایه تبدیل به ورودی لایه دوم کانولوشن تبدیل می شود. در مورد لایه اول، ورودی فقط صدای اصلی بود. با این حال ، هنگامی که در مورد لایه دوم متقاطع صحبت می شود ، ورودی نقشه (های) فعالسازی است که از لایه اول حاصل می شود. بنابراین هر لایه ورودی اساساً مکان هایی را در صدای اصلی توصیف می کند که در آن مشخصه های سطح پایین ظاهر می شوند. اکنون هنگامی که مجموعه ای از فیلترها روی آن اعمال می شود (و از لایه دوم جابجایی عبور داده می شود) ، خروجی فعال سازی هایی است که ویژگی های سطح بالاتری را نشان می دهند.

انواع این ویژگی ها می تواند قدرت صدا (بم . یا زیر بودن) یا طول موج صدا مانند (کوتاه و بلند) باشد. همانطور که از شبکه عبور می کنید و لایه های کانولوشن بیشتری را طی می کنید ، نقشه های فعال سازی دریافت می شوند که ویژگی های پیچیده تر و پیچیده تری را نشان می دهد. با عمیق تر شدن در شبکه ، فیلترها دارای یک میدان پذیرش بزرگتر و بزرگتر می شوند ، به این معنی که آنها می توانند اطلاعات را از یک منطقه بزرگتر از حجم ورودی اصلی را در نظر بگیرند. این است که آنها به منطقه بزرگتری از فضای داریه های صدا پاسخ می دهند.

لایه کاملاً متصل اساساً یک حجم ورودی می گیرد (خروجی کنولوشن یا ReLU یا لایه pool قبل از آن) و یک بردار ابعادی N را خروجی می دهد که N تعداد کلاس هایی است که برنامه باید از بین آنها انتخاب کند. برای مثال ، اگر برنامه طبقه بندی قطعه صدا از حیث احساسات باشد، N=5 خواهد بود زیرا 5 کلاس احساسات اعم از شادی، غم، ترس، عصبانیت و آرامش وجود دارد. هر عدد در این بردار ابعادی N نشان دهنده احتمال یک کلاس خاص است. نحوه عملکرد این لایه کاملاً متصل این است که به خروجی لایه قبلی (که باید نقشه های فعال سازی ویژگی های سطح بالا را نشان دهد) نگاه می کند و مشخص می کند که کدام ویژگی ها بیشتر با یک کلاس خاص مرتبط هستند، مثلا قطعه صدای حاوی قدرت صدای بالا است که ویژگی عصبانیت را متمایز می نمایاند. به عنوان مثال ، اگر برنامه پیش بینی کند که برخی از صداها عصبانی هستند ، در نقشه های فعال سازی که نشان دهنده ویژگی های سطح بالا مانند قدرت صدا و غیره که نمادهای عصبانیت بودن است ، مقادیر بالایی خواهد داشت.

به طور مشابه ، اگر برنامه پیش بینی کند که برخی از صداها با آرامش هستند ، در نقشه های فعال سازی که نشان دهنده ویژگی های سطح بالا مانند آرام بودن قدرت صوت و آوا و غیره که نمادهای آرامش است ، مقادیر بالایی خواهد داشت. اساساً ، یک لایه FC به ویژگیهای سطح بالا که بیشترین ارتباط را با یک کلاس

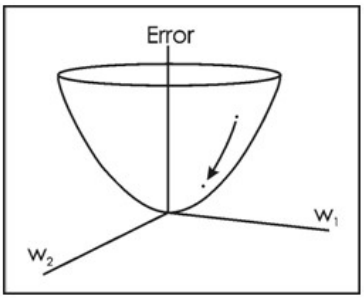
خاص دارد و وزنهای خاصی دارد نگاه می کند ، به طوری که وقت محاسبه خروجی ها بین وزنها و لایه قبلی ، احتمالات صحیح برای کلاسهای مختلف را بدست می آورید.

در حال حاضر ، این یکی از جنبه های شبکه های عصبی این است که چگونه فیلترهای اولین لایه کانولوشن می دانند که به دنبال قدرت صدا و یا طول موج ها هستند؟ چگونه لایه کاملاً متصلFC می داند که باید به کدام نقشه های فعال سازی نگاه کرد؟ چگونه فیلترهای هر لایه می دانند چه مقادیری باید داشته باشند؟ روشی که کامپیوتر می تواند مقادیر فیلتر (یا وزن) خود را تنظیم کند از طریق یک فرآیند آموزشی به نام انتشار مجدد عقبگردBack Propagation است.

وزنها یا مقادیر فیلتر تصادفی می شوند. فیلترها نمی دانند که به دنبال قدرت و طول موج هستند. فیلترهای لایه های بالاتر نمی دانند که باید به دنبال چه احساساتی باشند. با افزایش دوره های آموزشی epoch ها و گذر زمان صداهای متفاوتی به مدل داده شده است و برچسب مربوطه را به آنها الصاق می گردد. این ایده برای دادن صداهای متفاوت و برچسب ، فرآیند آموزشی است که CNN ها طی می کنند. یک مجموعه آموزشی وجود دارد که دارای هزاران صدا از احساسات متفاوت اعم از شادی، غم، ترس، عصبانیت و آرامش است و هر یک از صداها دارای برچسب است که آن صدا چه احساساتی دارد.
بنابراین انتشار عقب Back Propagation را می توان به 4 بخش مجزا ، دریافت صدا و آموزش مدل رو به جلو ، تخمین ضرر ، محاسبات مجدد وزن ها در پیمایش نورونها در انتشار عقبگرد و ارتقاء وزن تقسیم کرد. در طول انتشار رو به جلو ، یک صدای آموزشی گرفته می شود که یک آرایه m * n از اعداد است و آن را از کل شبکه عبور می دهد. از آنجا که همه وزنها یا مقادیر فیلتر به طور تصادفی راه اندازی شدند ، خروجی احتمالاً چیزی شبیه [.1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1] خواهد بود. خروجی که به هیچ عدد خاصی ترجیح داده نمی شود و این شبکه ، با وزن فعلی خود ، نمی تواند به دنبال ویژگی های سطح پایین باشد بنابراین نمی تولند نتیجه گیری منطقی در مورد طبقه بندی داشته باشد. این به بخش تابع از دست دادن back propagation می رود. آنچه در حال حاضر از آن استفاده می شود ، داده های آموزشی است. این داده ها دارای یک صدا و یک برچسب هستند. یک تابع محاسبه اشتباه را می توان به روش های مختلف تعریف کرد ، اما یک تابع رایج MSE (خطای میانگین مربع) است که برابر مربع (واقعی - مقدار پیش بینی شده)½ است.

$$Cost\ Function = \sum \frac{1}{2}\ (target - output)^2$$

مقدار اولیه تابع اشتباه برای اولین صدای آموزشی بسیار زیاد خواهد بود. مقصود رسیدن به جایی است که برچسب پیش بینی شده (خروجی ConvNet) با برچسب آموزشی یکسان باشد (این بدان معناست که شبکه پیش بینی خود را به درستی انجام داده است). برای رسیدن به آنجا ، می خواهیم میزان ضرر و اشتباه به حداقل برسد. با تصور این مسئله فقط به عنوان یک مشکل بهینه سازی در محاسبات ، یافت می شود که کدام ورودی ها (وزن ها) به طور مستقیم به (خطا) شبکه کمک کرده اند.



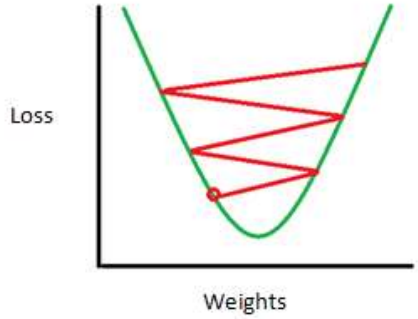
شکل (۵-۲) تصویر نشاندهنده یافتن مینیمم خطا است.

این معادل ریاضی dL/dW است که در آن W وزن یک لایه خاص است. در حال حاضر ، هدف این است که از شبکه به عقب حرکت کنیم ، که تعیین می کند کدام وزنه ها بیشترین ضرر یا خطا را داشته اند و راه هایی برای تنظیم آنها به گونه ای پیدا می شود که از دست دادن کاهش یابد.

$$w = w_i - \eta \frac{dL}{dW}$$

w = Weight
w_i = Initial Weight
η = Learning Rate

هنگامی که این مشتق محاسبه می شود، مدل آموزشی به آخرین مرحله که بروزرسانی وزن است می رود. اینجاست که همه وزن فیلترها را گرفته و آنها را به روز می کند تا در جهت مخالف گرادیان تغییر کنند. میزان یادگیری Learning Rate پارامتری است که توسط برنامه نویس انتخاب می شود. نرخ یادگیری بالا به این معنی است که گام های بزرگتری در به روزرسانی وزن برداشته می شود و بنابراین ، ممکن است زمان کمتری طول بکشد تا مدل در یک مجموعه بهینه از وزنه ها همگرا شود. با این حال ، میزان یادگیری بیش از حد بالا می تواند منجر به جهش هایی شود که بسیار بزرگ هستند و به اندازه کافی دقیق نیستند تا به نقطه مطلوب برسند.



شکل (۶-۲) تصویر نشاندهنده نرخ یادگیری است.

فرایند انتشار به جلو ، محاسبه خطا در وزنها، انتشار به عقب و به روزرسانی پارامترها یک تکرار آموزشی است. برنامه این فرایند را برای تعداد تکرار ثابت برای هر مجموعه از صدای آموزشی (که معمولاً دسته ای نامیده می شود) تکرار می کند. پس از اتمام به روزرسانی پارامتر در آخرین مثال آموزشی ، امید است شبکه به اندازه کافی آموزش ببیند تا وزن لایه ها به درستی تنظیم شود.

۳-۲- کتابخانه کراس (Keras)

کراس یک چهارچوب سطح بالا یادگیری عمیق پایتونی است که توسط آقای François Chollet در سال 2015 تأسیس شده. کراس چهارچوبی است که با آن و تنها با چند خط کد می‌توانیم برای ساختن شبکه‌های عصبی استفاده کنیم. البته کراس همه این کارها را خودش به تنهایی انجام نمی‌دهد، در حقیقت کراس یک فرانت‌اند (front-end) برای فریمورک های یادگیری عمیق تنسرفلو، CNTK و (theano) تیانو است و آن‌ها زیر ساخت شبکه‌های عصبی را می‌سازند و آموزش می‌دهند و برای همین به آن یک چهارچوب سطح بالا می گوییم چون کراس پیچیدگی استفاده از این کتابخانه‌ها را تا حد خوبی حذف می‌کند. یک ویژگی خاص دیگر کراس این است که محدود به یک کتابخانه یادگیری عمیق نیست و همانطور که گفتیم می‌توانیم از تنسرفلو، CNTK و یا تیانو برای محاسبات پشت پرده آن استفاده کنیم.

شکل (۷-۲) کراس یک کتابخانه سطح بالا برای تیانو، تنسورفلو، می باشد.

یک دلیل این محبوبیت کراس این بود که کراس انعطاف پذیری زیادی در استفاده از فریمورک های سطح پائین یادگیری عمیق محبوبی مثل تنسرفلو و CNTK دارد که تجربه کاربری فوق العاده و ساده ای را ارایه می‌دهد و لازم نیست که نگران برخی از جزئیات وقت گیر بود. علاوه بر این، مزیت دیگر کراس این است که به طور وسیعی هم توسط افراد آکادمیک و هم شرکت‌ها استفاده می‌شود و جامعه توسعه دهنده‌های آن هم پویا و بزرگ است. به علاوه چون کراس با پلتفرم‌های مختلفی سازگار است گزینه‌های بیشتری هم در اختیار داریم. به طور مثال، هم می‌توانیم کراس را بر روی سخت افزارهای مختلف مثل CPU، GPU و TPU (سخت افزار مخصوص یادگیری عمیق گوگل) و حتی سیستم‌های عامل تلفن همراه اجرا کنیم.

شاید بتوان یکی از مهم‌ترین ویژگی‌های کراس را در طراحی مدل‌های متنوع و از پیش تعیین شده کراس دانست. این مدلها بهترین رویه‌های (best practices) یادگیری عمیق در نظر گرفته شده‌اند و به صورت پیشفرض تنظیمات مورد استفاده در آنها اعمال شده است. بسیاری از مدل‌های پیشفرض کراس دارای بهترین تنظیمات مثل توابع فعال سازی و اندازه دسته که معمولاً در اغلب موارد نتایج خوبی می‌دهند، هستند. به علاوه در کراس مجموعه ای از مدل‌های از قبل آموزش داده شده مثل مدل ResNet50 که بر روی دیتاست ImageNet آموزش داده شده است وجود دارند که کار را برای انتقال یادگیری به مراتب ساده تر می‌کنند.

با این وجود همیشه استفاده از کراس شاید بهترین گزینه نباشد و باید دانست چه زمانی باید و چه زمانی نباید از آن استفاده کرد. وقتی با موارد زیر رو مواجه هستیم بهتر است به جای کراس از یک فریمورک سطح پائین تر مثل تنسرفلواستفاده کنیم.

- اگر هدف ایجاد یک سیستم با مقیاس پذیری بالا و برای پشتیبانی از تعداد زیادی کاربر است.

- اگر محدودیت‌هایی که در حافظه و قدرت پردازشی وجود دارد از محدودیتی که صرف زمان کدنویسی قرار است تلف شود مهم تر باشند.

۲-۴- مزایای کتابخانه keras در پایتون

- مدل های آماده
- پشتیبانب توسط شرکت های بزرگ نظیر Google, Microsoft, Amazon, Apple, Nvidia, Uber
- منعطف و قابل تغییر
- قابلیت اجرا در پلتفرم های iOS, Android,web API
- سریع بودن
- به وضوح بیان کردن خطاها

۱-۴-۲- کتابخانه ی کراس چیست؟

کراس کتابخانه ای است که با آن و تنها با چند خط کد می‌توانیم برای ساختن شبکه‌های عصبی استفاده کنیم. البته کراس همه این کارها را خودش به تنهایی انجام نمی‌دهد، در حقیقت کراس یک فرانت‌اند(front-end) برای فریمورک های یادگیری عمیق تنسرفلو، CNTKو (مرحوم) تیانو است و آن‌ها پشت شبکه‌های عصبی را می‌سازند و آموزش می‌دهند و برای همین به آن یک چهارچوب سطح بالا می گوییم چون کراس پیچیدگی استفاده از این کتابخانه‌ها را تا حد خوبی حذف می‌کند. یک ویژگی خاص دیگر کراس این است که محدود به یک کتابخانه یادگیری عمیق نیست و همانطور که گفتیم می‌توانیم از تنسرفلو، CNTKو یا تیانو برای محاسبات پشت پرده آن استفاده کنیم Keras .یک کتابخانه یادگیری عمیق برای آموزش سریع و کارآمد مدل های یادگیری عمیق است و همچنین می تواند با Tensorflow و Theano کار کند. از آنجا که سبک وزن و بسیار آسان برای استفاده است، Kerasدر یک زمان بسیار کم محبوبیت زیادی به دست آورده است. ساده ترین کتابخانه در این حوزه را می‌توان ار پایتون کراس **(Keras)** نام برد. کراس از رنج گسترده ایی از شبکه های عصبی پشتیبانی می‌کند و ساختن نمونه های اولیه را بسیار ساده می‌کند. و از همه مهمتر تحلیل کد آن هم بسیار ساده است. البته به عنوان یکی از نقاط قوت آن می‌توان به این نکته اشاره کرد که این کتابخانه از چند **GPU** پشتیبانی می‌کند Keras .یک شبکه عصبی با سطح بالا است که به زبان Python نوشته شده و قادر به اجرا در بالای TensorFlow ، CNTKیا Theano است. این برنامه با تمرکز بر فعال کردن سریع آزمایش انجام شد. توانایی رفتن از ایده به نتیجه با حداقل تاخیر ممکن برای انجام تحقیقات خوب مهم است Keras .بهترین اقدامات را برای کاهش بار محسباتی دنبال می کند API :های سازگار و ساده را ارائه می دهد ، تعداد اقدامات کاربر مورد نیاز برای موارد استفاده معمولی را به حداقل می رساند و بازخورد واضح و عملی را در مورد خطای کاربر فراهم می کند. به طور خاص ، لایه های عصبی ، توابع هزینه ، بهینه سازها

، برنامه های اولیه سازی ، توابع فعال سازی و برنامه های منظم سازی ، همه ماژول های مستقل هستند که می توانید برای ایجاد مدل های جدید در کراس ترکیب کنید Keras .با زبانهای یادگیری عمیق سطح پایین تر) به ویژه (TensorFlow ادغام می شود ، این امکان را برای شما فراهم می کند تا بتوانید هر چیزی را که می توانستید به زبان پایه ساخته باشید ، پیاده سازی کنید. به طور خاص ، به عنوان tf.keras ، API Keras یکپارچه با گردش کار TensorFlow شما ادغام می شود.

۲-۴-۲- دلیل استفاده از کراس

کراس این امکان را به کاربران می دهد تا بتوانند گراف ها را بصورت پویا فراهم نمایند همچنین کراس ابزار ها و قابلیت های خوبی را برای کاربران خود فراهم نموده است تا بتواند نسبت به کار هایی که به آن نسبت می دهند انعطاف پذیر بوده و کار ها را با سرعت بیشتری به پایان برساند.

- ❖ آسان نسبت به اغلب فریمورک ها
- ❖ تحلیل کد آن هم ساده
- ❖ قابلیت استفاده از چند GPU
- ❖ ماژولاریتی بالا
- ❖ انعطاف پذیری بالا
- ❖ پشتیبانی همزمان از چندین backend
- ❖ قابلیت ایجاد مدل های ترتیبی و تابعی
- ❖ دیتاست های آماده
- ❖ مدل های اماده زیاد
- ❖ اجرای همزمان روی چند GPU و چند سیستم

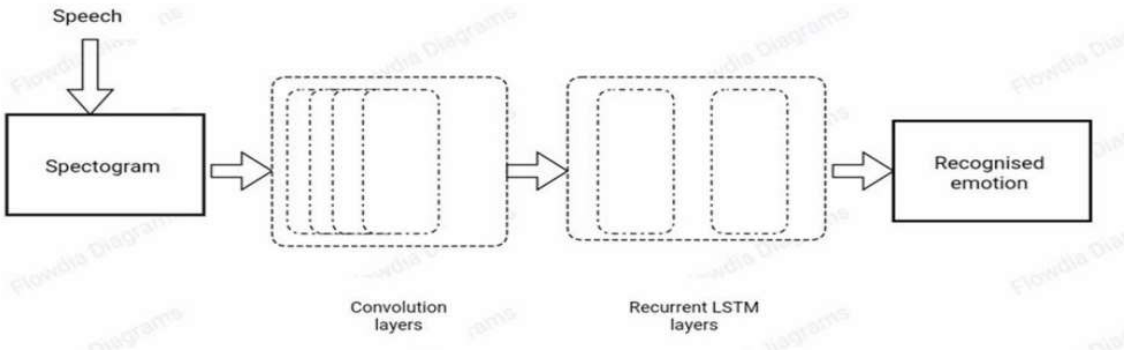
۲-۵- مروری بر منابع

۲-۵-۱- تکنیک ترکیبی توسط CNN+LSTM برای تشخیص احساسات گفتار [۲]

تشخیص خودکار احساسات گفتار یک فعالیت بسیار ضروری برای تعامل موثر انسان و کامپیوتر است. این مقاله با استفاده از طیف‌نگارها به عنوان ورودی به LSTM کانولوشنال عمیق ترکیبی برای تشخیص احساسات گفتار ایجاد شده است. در این مطالعه، مدل پیشنهادی خود را با استفاده از چهار لایه کانولوشن برای استخراج ویژگی سطح بالا از طیف‌نگارهای ورودی، لایه LSTM برای تجمع وابستگی‌های بلندمدت و در

نهایت دو لایه متراکم آموزش دادیم. نتایج تجربی در پایگاه داده SAVEE عملکرد امیدوارکننده ای را نشان می دهد. مدل پیشنهادی به دلیل به دست آوردن دقت 94.26 درصد توانایی بالایی دارد.

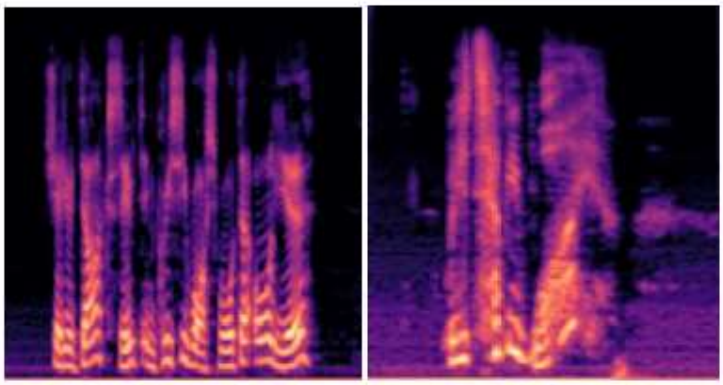
شکل ۸-۲ معماری پایه کار را نشان می دهد. این به طور گسترده شامل دو بخش است: (۱) ایجاد طیف‌نگارهای سیگنال گفتار، (۲) معماری شبکه: همجوشی.CNN-LSTM این ادغام بسیار مؤثر است زیرا از هر دو مزیت شبکه های عصبی کانولوشنال و LSTM استفاده می کند.



شکل (۸-۲) معماری مدل پیشنهادی مقاله توسط شبکه عصبی عمیق پیچشی و حافظه طولانی کوتاه مدت.

۲-۵-۱-۲-ایجاد طیف نگار

نمایش بصری صدا را طیف نگار می نامند. توسط یک الگوریتم ریاضی به نام تبدیل سریع فوریه ساخته شده است. یک سیگنال گفتاری خام گرفته شده و با استفاده از این الگوریتم به اجزای فرکانس آن تجزیه می شود. به بیان ساده، یک طیف نگاری تغییر فرکانس را منعکس می کند .در سیگنال یک طیف نگار زمان افقی محور x و فرکانس محور y عمودی را نشان می دهد. اجزاء (که یک سیگنال پیچیده را تشکیل می دهند) در سیگنال گفتاری مقدار دامنه یکسانی ندارند. تفاوت در دامنه در یک طیف گرا با سایه نشان داده می شود. به این ترتیب، یک طیف نگار سه بعدی است یعنی زمان محور x، فرکانس محور y و دامنه سایه مشابه RGB را نشان می دهد. طیف‌نگارهای نمونه در شکل 2 نشان داده شده‌اند. در مدل خود از 480 گفته استفاده کرده‌ایم که طیف‌نگارهای متناظر با استفاده از تابع طیف‌نگار پایتون از کتابخانه pyplot تولید می‌شوند.



شکل (۹-۲) طیف نگار تولید شده توسط سیگنالهای گفتاری.

۲-۵-۱-۳- معماری شبکه: CNN-LSTM Fusion

❖ Convolution neural network

CNN در حال حاضر در بسیاری از کاربردها از طبقه بندی تصاویر گرفته تا سنتز صدا استفاده می شود. معمولاً، باید شبکه عصبی کانولوشن را به عنوان یک شبکه عصبی مصنوعی در نظر گرفت که توانایی شناسایی الگوها و درک آنها را دارد. این تشخیص الگوی شبکه های عصبی را برای تجزیه و تحلیل تصویر بسیار مفید می کند. در CNN عمدتاً چهار لایه وجود دارد و اینجا از چند لایه اضافی برای عادی سازی شبکه استفاده خواهد شد. هر یک از این موارد در زیر توضیح داده شده است.

لایه پیچیدگی: لایه پیچیدگی نشان دهنده لایه یک CNN است که در آن ما (تصاویر، داده های سری زمانی 1 بعدی) با فیلترها یا هسته ها تعامل داریم. با استفاده از یک پنجره کشویی، واحدهای کوچکی را در سراسر ورودی اعمال می کنیم و این واحدها به عنوان فیلتر شناخته می شوند. عمق ورودی و فیلتر مترادف هستند، یک تصویر رنگی RGB با عمق سه، با همان عمق یعنی سه فیلتر می شود. در فرآیند کانولوشن، حاصلضرب المان فیلترها در تصویر گرفته می شود و سپس برای هر حرکت لغزشی محصولات اضافه می شوند. پس از انحراف یک فیلتر سه بعدی به عنوان خروجی، ماتریس دو بعدی را به دست خواهیم آورد.

لایه فعال سازی: بین لایه های کانولوشن متوالی، ما فقط از توابع فعال سازی غیر خطی استفاده می کنیم. با توجه به خاصیت انجمنی کانولوشن، فقط توابع فعال سازی غیرخطی بین لایه های کانولوشن متوالی مجاز هستند و توابع فعال سازی خطی منجر به یادگیری نمی شوند.

لایه ادغام: ادغام شامل نمونه برداری پایین از ویژگی ها با این هدف است که در طول آموزش، پارامترهای کمتری را یاد بگیریم. با استفاده از لایه ادغام، عمدتا دو پارامتر هایپر معرفی می شوند که یکی بعد وسعت فضایی و دیگری گام است. مقدار "n" بعد وسعت فضایی را تعریف می کند، با گرفتن n * n نمایش ویژگی و نگاشت به یک مقدار واحد. تعداد ویژگی هایی که پنجره کشویی در امتداد آن رد می شود .عرض و ارتفاع گام است. برازش بیش از حد با انجام ادغام کاهش می یابد زیرا تعداد پارامترها را کاهش می دهد. یک فیلتر 2 * 2 حداکثر بدون همپوشانی با گام 2 نشان دهنده یک لایه مشترک است. اگر یک مقدار حداکثر در بین ویژگی های منطقه برگردانده شود، یک فیلتر حداکثر را نشان می دهد، اما اگر بازگشت میانگین ویژگی ها باشد، فیلتر متوسط است. در عمل، فیلتر حداکثر عملکرد بهتری دارد.

لایه کاملا متصل: ویژگی های سطح بالا در داده ها با خروجی لایه کانولوشن نشان داده می شود. ما از این لایه برای طبقه بندی استفاده می کنیم. یک لایه کاملا متصل معرفی شده است تا اجازه دهد خروجی صاف شود و به لایه خروجی متصل شود تا این ویژگی ها در ترکیب های غیر خطی یاد بگیرند. خروجی لایه های ادغام حجم سه بعدی است اما یک شبکه فید فوروارد کاملا متصل یک بردار ویژگی 1 بعدی را به عنوان ورودی می گیرد. برای تبدیل این حجم سه بعدی به یک بعد، عرض و ارتفاع خروجی باید یک باشد و این تنها با صاف کردن لایه سه بعدی به وکتور یک بعدی امکان پذیر است.

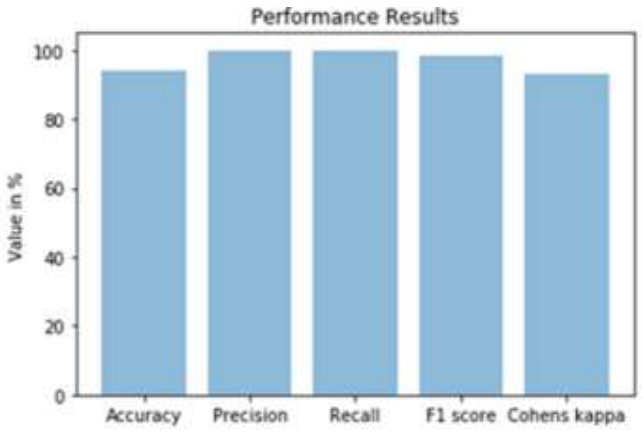
لایه Normalization Batch (Batch Norm): در حین آموزش، اگر در هر لایه از شبکه عصبی ما ناپایداری وجود داشته باشد، نرمال سازی دسته ای را روی آن لایه اعمال می کنیم. خروجی از تابع فعال سازی با استفاده از یک لایه عادی سازی دسته ای نرمال می شود و این اولین کاری است که این لایه انجام می دهد. این اضافه می تواند سرعت تمرین را تا حد زیادی افزایش دهد. همچنین وزنه‌های بزرگ دور تا حد زیادی بر روند تمرین تأثیر می‌گذارد و هنجار دسته‌ای آن را کاهش می‌دهد. نام هنجار دسته ای داده شده است زیرا بر اساس هر دسته کار می کند و اندازه دسته زمانی تنظیم می شود که مدل خود را آموزش می دهیم.

❖ حافظه بلند مدت کوتاه مدت (LSTM)

RNN (شبکه عصبی بازگشتی) یک شبکه عصبی است که در آن خروجی مراحل قبلی به عنوان ورودی به مرحله فعلی تغذیه می شود. اما در عمل، این شبکه‌های عصبی مکرر محدودیتی دارند که می‌توانند تنها چند قدم به عقب نگاه کنند. برای درک مشکلاتی مانند تشخیص گفتار، یک سیستم برای ذخیره و استفاده از اطلاعات زمینه مورد نیاز است. در واقع LSTM یک نوع RNN است. یک شبکه عصبی که وابستگی های نظم را در مسائل پیش بینی توالی یاد می گیرد. شبکه های NN به منظور اصلاح مشکل اضمحلال گرادیان، تلاش های بی پایانی به کار گرفته شده و LSTM یکی است. سیگنال گفتار در حوزه زمان پیوسته است، به طوری که هر تابع فریم فقط ویژگی های احساسی را در یک فریم واحد نشان می دهد. LSTM اطلاعات بین فریم های مجاور را افزایش می دهد که به بازتاب تداوم زمانی ویژگی ها کمک می کند. بنابراین، LSTM آشکارا از تشخیص گفتار پشتیبانی می کند.

❖ ادغام CNN-LSTM

یک شبکه عصبی کانولوشن که یک شبکه پیشخور است، داده‌های مکانی را فیلتر می‌کند در حالی که شبکه عصبی مکرر (LSTM) داده‌ها را به خود باز می‌گرداند. بنابراین شبکه های عصبی مکرر برای داده های متوالی مناسب تر هستند. به عبارت دیگر، یک شبکه عصبی کانولوشن قادر به درک الگوها در سراسر فضا است، LSTM می تواند آنها را در طول زمان ببیند. از آنجایی که سیگنال گفتار ما متوالی است، بنابراین LSTM بهترین گزینه برای پردازش گفتار است. این مدل در پایتون ساخته شده و برای 100 دوره آموزش داده شده است. دقت، یادآوری، دقت، امتیاز F1 پارامترهایی هستند که تجزیه و تحلیل عملکرد مدل بر اساس آنها نشان داده شده است. نمودار زیر نتایج را بر اساس این پارامترها نشان می دهد. پس از نمودار عملکرد، تصاویری از طیف‌نگارهای تولید شده برای هر سیگنال صوتی، نتایج برنامه خروجی و احساسات شناسایی شده برای یک ورودی صوتی خاص را نشان می‌دهد.



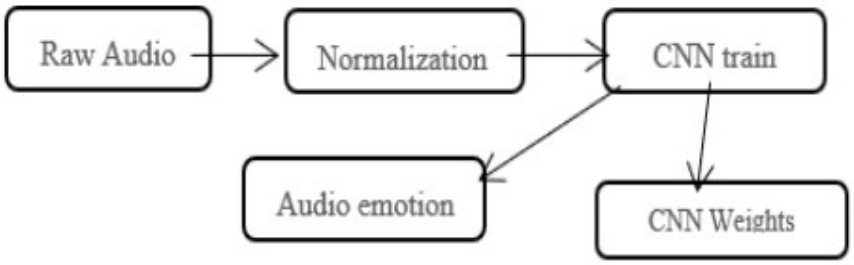
شکل (۱۰-۲) نتایج عملکرد مدل بر اساس پارامترهای مختلف.

۲-۵-۲- تشخیص احساسات گفتاری توسط شبکه عصبی عمیق پیچشی [۳]

افراد مختلف احساسات متفاوتی دارند و در مجموع روش متفاوتی برای ابراز آن دارند. عواطف گفتاری انرژی های متفاوتی دارند، در صورت در نظر گرفتن موضوعات مختلف، بر تغییرات زیر و بمی تأکید می شود. بنابراین، تشخیص عواطف گفتاری یک کار طاقت فرسا در بینایی محاسباتی است. در اینجا، تشخیص احساسات گفتار بر اساس الگوریتم شبکه عصبی کانولوشن (CNN) است که از ماژول‌های مختلفی برای تشخیص احساسات استفاده می‌کند و طبقه‌بندی‌کننده‌ها برای تمایز احساساتی مانند شادی، تعجب، خشم، حالت خنثی، غم و غیره استفاده می‌شوند. مجموعه داده برای سیستم تشخیص احساسات گفتار، نمونه های گفتاری است و ویژگی ها با استفاده از بسته LIBROSA از این نمونه های گفتاری استخراج شده است. عملکرد طبقه بندی بر اساس ویژگی های استخراج شده است. در نهایت می توانیم احساس سیگنال گفتار را تعیین کنیم.

یادگیری عمیق در یک اصطلاح واحد می تواند به عنوان سیستم عصبی انسان درک شود. مجموعه‌های یادگیری عمیق ماشین بینایی برای یادگیری از طریق مجموعه‌ای از صدا/تصویر که به عنوان داده‌های آموزشی نیز شناخته می‌شوند، ساخته شده‌اند تا مشکل را برطرف کنند. مدل‌های مختلف یادگیری عمیق، یک کامپیوتر را آموزش می‌دهد تا مانند یک انسان تجسم کند. مدل‌های یادگیری عمیق بر اساس ورودی‌های گره‌ها می‌توانند تجسم کنند. از این رو نوع شبکه مانند سیستم عصبی انسان است که هر گره

تحت یک شبکه بزرگتر به عنوان یک نورون عمل می کند. بنابراین، مدل های یادگیری عمیق اساسا بخشی از شبکه های عصبی مصنوعی هستند. الگوریتم های یادگیری عمیق به طور عمیق در مورد ورودی یاد می گیرد Tصدا/تصویر هنگام عبور از هر لایه شبکه عصبی. ویژگی‌های سطح پایین مانند لبه‌ها با یادگیری داده‌شده به لایه‌های اولیه شناسایی می‌شوند و لایه‌های متوالی ویژگی‌های لایه‌های قبلی را در یک نمایش فلسفی‌تر با یکدیگر همکاری می‌کنند. تصاویر، صداها، داده‌های سانسور و سایر داده‌ها الگوهای اشکال دیجیتالی هستند که یادگیری عمیق آنها را تشخیص می‌دهد. برای پیش‌بینی، داده‌ها را از قبل آموزش می‌دهیم و یک مجموعه آموزشی و مجموعه آزمایشی می‌سازیم (نتایج مشخص است). همانطور که پیش بینی ما یک گره بهینه به دست می آورد به طوری که گره پیش بینی شده خروجی رضایت بخشی را ارائه می دهد.



شکل (۱۱-۲) متدولوژی آنالیز سیگنالهای گفتاری.

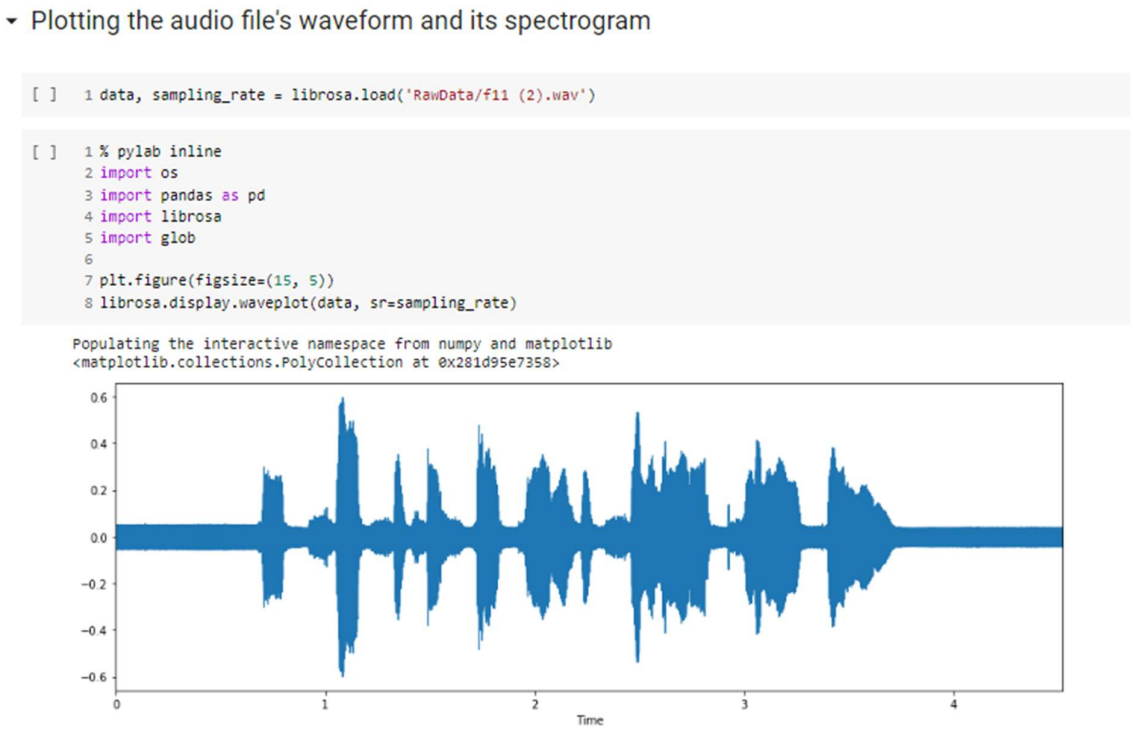
یک داده آموزشی به سیستم واکشی می شود که شامل برچسب عبارت است و آموزش وزن نیز برای آن شبکه ارائه می شود. یک صدا به عنوان ورودی گرفته می شود. پس از آن، نرمال سازی شدت روی صدا اعمال می شود. یک صدای عادی برای آموزش شبکه Convolutional استفاده می شود، این کار برای اطمینان از اینکه تأثیر دنباله ارائه مثال ها بر عملکرد آموزش تأثیر نمی گذارد انجام می شود. مجموعه ای از وزنه ها به عنوان یک نتیجه از این فرآیند آموزشی بیرون می آید و بهترین نتایج را با این داده های یادگیری به دست می آورد. در حین آزمایش، مجموعه داده سیستم را با گام و انرژی دریافت می‌کند و بر اساس وزن‌های شبکه نهایی آموزش داده شده، احساسات تعیین‌شده را نشان می‌دهد. خروجی در یک مقدار عددی نشان داده می شود که هر کدام مربوط به یکی از پنج عبارت است. 3 احساس وجود دارد که بر اساس مقدار ضربان در دقیقه شخص تشخیص داده می شود، آن ها آرامش / آرامش، شادی / سرگرمی، ترس / خشم. رنگ ها و اشکال هنر تولید شده موازی با احساسات کشف شده بر اساس اصول "روانشناسی رنگ" و "روانشناسی شکل" است. پس از ساخت مدل‌های مختلف، مدل CNN بهتری را برای کار تمایز احساسات دریافت کردیم. ما به دقت 71 درصد از مدل موجود قبلی رسیدیم. مدل ما با داده های بیشتر بهتر عمل می کرد. همچنین مدل ما هنگام تشخیص صدای مردانه و زنانه بسیار خوب عمل کرد.

فصل ۳:

روش تحقیق

۳-۱- مقدمه

سیگنال صوتی یک سیگنال سه بعدی است که در آن 3 محور زمان، دامنه و فرکانس را نشان می دهد. ما از librosa برای تجزیه و تحلیل و استخراج ویژگی های هر سیگنال صوتی استفاده خواهیم کرد. تابع (load.) یک فایل صوتی را می کشد و آن را در یک آرایه 1 بعدی که از سری زمانی x است رمزگشایی می کند و SR در واقع نرخ نمونه برداری x است. SR به طور پیش فرض 22 کیلوهرتز است. در اینجا من یک نمایش فایل صوتی را با استفاده از تابع (IPython.display) نشان خواهم داد. Librosa.display برای نمایش فایل های صوتی در اشکال مختلف مانند نمودار موج، طیف نگار و نقشه رنگی مهم است.

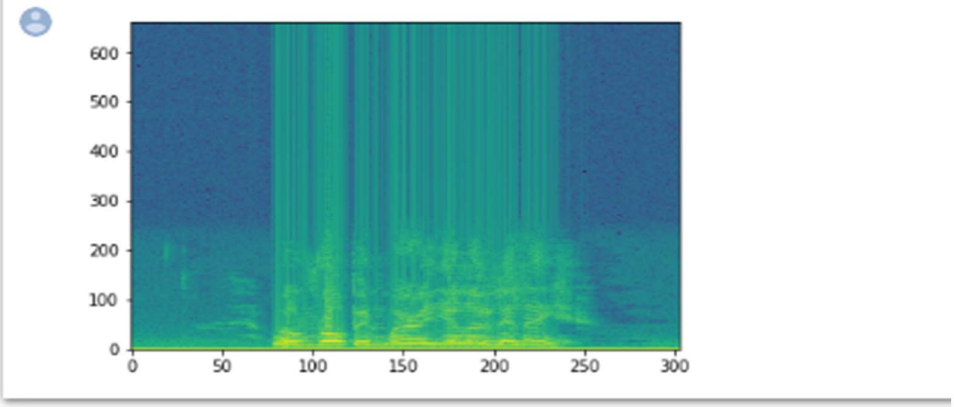


شکل (۱-۳) نمایش فایل های صوتی در اشکال مختلف مانند نمودار موج، طیف نگار.

نمودارهای موج از بلندی صدا در یک زمان خاص استفاده می کنند. Spectrogram فرکانس های مختلف را برای یک زمان خاص با دامنه خود نمایش می دهد. برای آموزش مدل برای محاسبه دقت. (module02) در این ماژول ما مدل را برای تخمین دقت آموزش می دهیم. اول، ماژول های لازم را وارد کنید. سپس مجموعه داده را بکشید. ما مقدار نرخ نمونه برداری را با بسته های librosa و تابع mfcc دریافت خواهیم کرد. پس از آن این مقدار متغیرهای دیگر را در خود جای داده است. حالا فایل‌های صوتی و مقدار mfcc متغیری

را نگه می‌دارند، در نتیجه یک لیست اضافه می‌کند. سپس لیست را فشرده کنید و دو متغیر x & y را نگه دارید. سپس مقادیر شکل (x, y) را با استفاده از بسته numpy نمایش داده ایم.

```
import matplotlib.pyplot as plt
import scipy.io.wavfile
import numpy as np
import sys


sr,x = scipy.io.wavfile.read('RawData/f10 (2).wav')

## Parameters: 10ms step, 30ms window
nstep = int(sr * 0.01)
nwin  = int(sr * 0.03)
nfft = nwin

window = np.hamming(nwin)

## will take windows x[n1:n2].  generate
## and loop over n2 such that all frames
## fit within the waveform
nn = range(nwin, len(x), nstep)

X = np.zeros( (len(nn), nfft//2) )

for i,n in enumerate(nn):
    xseg = x[n-nwin:n]
    z = np.fft.fft(window * xseg, nfft)
    X[i,:] = np.log(np.abs(z[:nfft//2]))

plt.imshow(X.T, interpolation='nearest',
    origin='lower',
    aspect='auto')

plt.show()
```



شکل (۲-۳) نمایش فایل های صوتی با استفادهد از Numpy, Fast Fourier Transform.

فرآیند پیاده سازی مدل CNN. (module03) گفتار در قالب تصویر با 3 لایه نمایش داده می شود. هنگام استفاده از CNN، مشتقات اول و دوم تصویر گفتار را با زمان و فرکانس در نظر بگیرید. CNN می تواند داده های گفتار را پیش بینی، تجزیه و تحلیل کند، CNN می تواند از سخنرانی ها بیاموزد و کلمات یا گفته ها را شناسایی کند. طبقه بندی احساسات گفتاری. (module04) هنگام آزمایش، ورودی صوتی را ارائه می دهیم. بعد، صدا را برای شنیدن با بسته‌های ipython.disply اجرا می‌کنیم. سپس ویژگی های صوتی را با بسته های librosa.display.waveplot ترسیم کنید. ویژگی ها را با استفاده از librosa.load استخراج کنید. این یک قاب داده را تبدیل می کند و فرم ساختار یافته را نمایش می دهد. علاوه بر این، مدل بارگذاری شده را با اندازه دسته‌ای تابع پیش‌بینی 32 مقایسه می‌کند. در نهایت خروجی فایل صوتی را نشان می‌دهد که آن فایل صوتی چه نوع بیان/احساساتی دارد.

در مدل CNN چهار لایه مهم وجود دارد:

1. لایه کانولوشنال: مناطق برجسته را در فواصل زمانی مشخص می کند، گفته های طولی که متغیر هستند و توالی نقشه ویژگی را به تصویر می کشد.

2. لایه فعال سازی: یک تابع لایه فعال سازی غیر خطی به طور معمول برای خروجی های لایه کانولوشن استفاده می شود. در این مورد ما از واحد خطی اصلاح شده (ReLU) در طول کار خود استفاده کرده ایم.

3. لایه Max Pooling: این لایه گزینه هایی را با حداکثر مقدار برای لایه های متراکم فعال می کند. این کمک می کند تا ورودی های طول متغیر را در یک آرایه ویژگی با اندازه ثابت نگه دارید.

4. استخراج و تجسم ویژگی های صوتی لایه متراکم. (module01) استخراج مشخصه ها برای طبقه بندی و ترسیم مورد نیاز است.

```
[ ]  1
     2 x_traincnn =np.expand_dims(X_train, axis=2)
     3 x_testcnn= np.expand_dims(X_test, axis=2)
```

```
     1 model = Sequential()
     2
     3 model.add(Conv1D(256, 5,padding='same',
     4                  input_shape=(216,1)))
     5 model.add(Activation('relu'))
     6 model.add(Conv1D(128, 5,padding='same'))
     7 model.add(Activation('relu'))
     8 model.add(Dropout(0.1))
     9 model.add(MaxPooling1D(pool_size=(8)))
    10 model.add(Conv1D(128, 5,padding='same',))
    11 model.add(Activation('relu'))
    12 #model.add(Conv1D(128, 5,padding='same',))
    13 #model.add(Activation('relu'))
    14 #model.add(Conv1D(128, 5,padding='same',))
    15 #model.add(Activation('relu'))
    16 #model.add(Dropout(0.2))
    17 model.add(Conv1D(128, 5,padding='same',))
    18 model.add(Activation('relu'))
    19 model.add(Flatten())
    20 model.add(Dense(10))
    21 model.add(Activation('softmax'))
    22 opt = keras.optimizers.rmsprop(lr=0.00001, decay=1e-6)
```

شکل (۳-۳) نمایش لایه های فعالسازی Activation، کانولوشنConvolution، Max Pooling، Dense

```
1 model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_9 (Conv1D)            (None, 216, 256)          1536
_________________________________________________________________
activation_11 (Activation)   (None, 216, 256)          0
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 216, 128)          163968
_________________________________________________________________
activation_12 (Activation)   (None, 216, 128)          0
_________________________________________________________________
dropout_4 (Dropout)          (None, 216, 128)          0
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 27, 128)           0
_________________________________________________________________
conv1d_11 (Conv1D)           (None, 27, 128)           82048
_________________________________________________________________
activation_13 (Activation)   (None, 27, 128)           0
_________________________________________________________________
conv1d_12 (Conv1D)           (None, 27, 128)           82048
_________________________________________________________________
activation_14 (Activation)   (None, 27, 128)           0
_________________________________________________________________
flatten_3 (Flatten)          (None, 3456)              0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                34570
_________________________________________________________________
activation_15 (Activation)   (None, 10)                0
=================================================================
Total params: 364,170
Trainable params: 364,170
Non-trainable params: 0
_________________________________________________________________
```

شکل (۳-۴) نمایش خلاصه مدل.

۲-۳- دیـتـا ست

مجموعه آموزشی داده شده بصورت زیر است:

$$S = \{ x^{(i)}, y^{(i)} \} \qquad m\ audios\ x^{(i)}\ is\ i^{th}\ audio$$

$$y^{(i)} \in \{1, 2, 3, 4, 5\}$$

$$If\ y^{(i)} = 1 \Rightarrow x^{(i)}\ is\ Sad$$

$$If\ y^{(i)} = 2 \Rightarrow x^{(i)}\ is\ Happy$$

$$If\ y^{(i)} = 3 \Rightarrow x^{(i)}\ is\ Angry$$

$$If\ y^{(i)} = 4 \Rightarrow x^{(i)}\ is\ Calm$$

$$If\ y^{(i)} = 5 \Rightarrow x^{(i)}\ is\ Fear$$

در مجموعه آموزشی تعداد 20000= i قطعه صدا یا احساسات متفاوت داریم. این صدا ها که به ۲۵ نفر تعلق دارند هر کدام در پوشه مخصوصی با نام (Actor_01 … Actor_25) دسته بندی می شوند، در نتیجه برچسب آنها براساس نوع احساسات نامگذاری می شود.

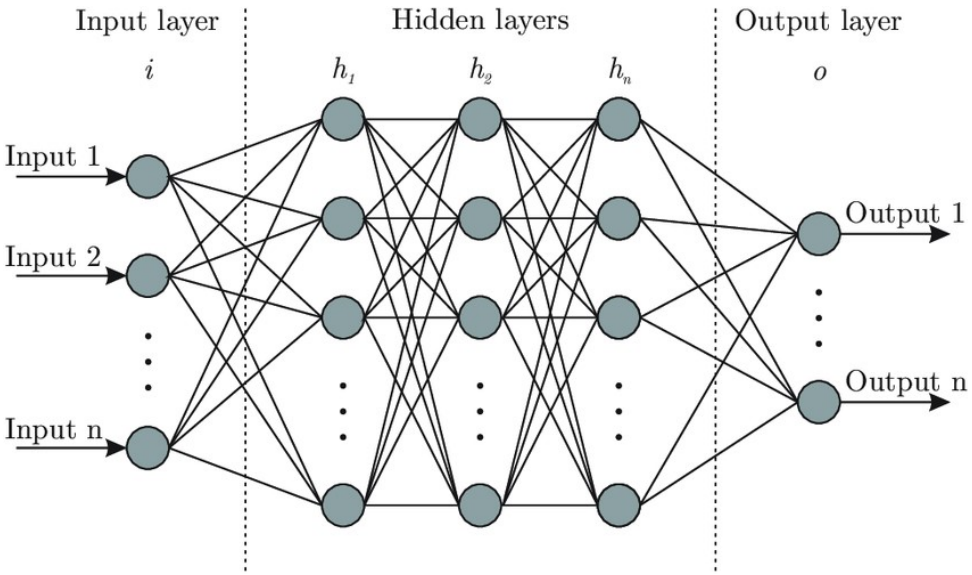
Importing the required libraries

```
import librosa
import librosa.display
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from matplotlib.pyplot import specgram
import keras
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.layers import LSTM
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
from keras.layers import Input, Flatten, Dropout, Activation
from keras.layers import Conv1D, MaxPooling1D, AveragePooling1D
from keras.models import Model
from keras.callbacks import ModelCheckpoint
from sklearn.metrics import confusion_matrix
```

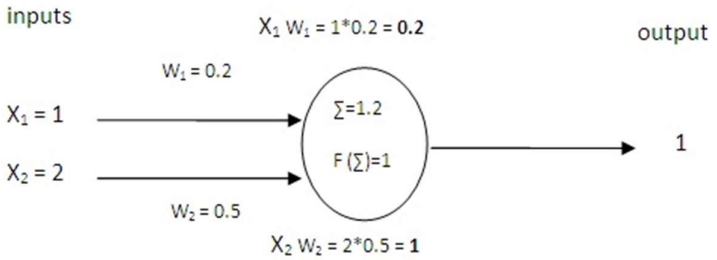
شکل (۵-۳) وارد نمودن کتابخانه ها

شبکه های عصبی مصنوعی را به نوت بوک معرفی می کنیم. همانطور که در شکل زیر ۶-۳ مشاهده می شود شبکه های عصبی دارای سه قسمت ورودی؛ مخفی؛ و خروجی است. همانطور که پیش تر گفته شد اصوات در سیستم به شکل آرایه ای از اعداد می باشند که رویهم یک ماتریس را تشکیل می دهند. این ورودیها در لایه اول شبکه عصبی وارد می شوند. هر درایه از این المانهای ورودی مبین یک داده در پیکسلهای تصویر می باشند که دارای وزنی جهت نمایش اهمیت و شاخصی برای موثر بودن آن نود Node در محاسبات می باشند. در تصویر تک نورون دو ورودی X1=1 , X2=2 هر کدام دارای وزن W1=0.2, W2=0.5 هستند که وزن X2 با ۰.۵ بیش از دو برابر X1 است و در نتیجه سنگین تر و با اهمیت تر است.
برای مثال چنانچه یک راه برای کاهش ابعاد و سودجستن از ابعاد مهم است؛ خاموشی نورونهایی است که وزن کنری دارند در نتیجه بهر بردن از نورونها با وزن بالاتر کمک به حذف داده های اضافی می نماید که باعث رشد سرعت در محاسبات می شود، علاوه بر آن همچنین تمرکز بر داده های با اهمیت بالا و مفید نرخ دقت در پیش بینی را بالا می برد.
در شبکه های عصبی هر چقدر تعداد لایه های مخفی بیشتر باشند دال بر وجود داده های بیشتر می باشند، این امر مانند آن است که یک تصویر با وضوح بالا داده ها و یا یک فایل صوتی با شفافیت بالا و لایه های مخفی بیشتر دارد و پیش بینی بر اساس این حجم از داده بسیار راحتتر و دقیق تر می باشد. تصور اینکه

تصاویر زیادی با وضوح بالا در اختیار است می تواند دقت در پیش بینی را تا حد زیادی بالا ببرد اما در عین حال پیچیدگی محاسبات بالا رفته و نیازمند به استفاده از پروسسورهایی با توان عملیاتی بالا برای مدیریت محاسبات سنگین می باشد.



شکل (۶-۳) شبکه عصبی مصنوعی با لایه های متعدد مخفی



شکل (۷-۳) تک نورون عصبی همراه دو ورودی و وزنهایش

در خط بیست و دوم شکل ۳-۳ کتابخانه کراس اپتیمایزر را در نوت بوک استفاده می کند. این کتابخانه یک پکیج پیاده سازی از الگوریتم های مختلف بهینه سازی است مانند:

گرادیان کاهشی تصادفی

معمولا پس از طراحی مدل و لرنینگ آن با داده های آموزشی، نوبت به یافتن بهترین جوابها می رسد. صد البته در طراحی انجام شده شاهد خطاهایی خواهیم بود و این بدین معنی است که تا رسیدن به بهترین

نقطه که جواب آرمانی است و کمترین خطا را دارا می باشد، می بایست ابتدا میزان خطا را بااستفاده از تابع هزینه (مقدار واقعی – مقدار پیش بینی شده)محاسبه نمود و سپس آنچه در ریاضیات رسم است از این تابع مشتق گیری ضمنی بر اساس پارامترهایی مانند(عرض از مبدا و یا شیب) گرفت. در نهایت پارامترهای جدید را مقدار دهی نموده، با تفریق مقدار قدیمی از ضرب لرنینگ ریت یا نرخ یادگیری در مجموع مشتقات ضمنی همان پارامتر و مجددا مدل را در تکرار بعدی آموزش داده و دوباره با استفاده از تابع هزینه، خطا را محاسبه می نماییم. تفاوت گرادیان کاهشی تصادفی با گرادیان کاهشی استاندارد در این است که برخلاف گرادیان کاهشی استاندارد که برای بهینه‌سازی تابع هدف از تمام داده‌های آموزشی استفاده می‌کند، گرادیان کاهشی تصادفی از گروهی از داده‌های آموزشی که به‌طور تصادفی انتخاب می‌شود برای بهینه‌سازی استفاده می‌کند. این روش در مسائل آماری و یادگیری ماشین کاربرد فراوانی دارد.

$$Sum\ of\ Squared\ Error\ (SSE) = \frac{1}{2}\ sum(Y_{\text{واقعی}} - Y_{\text{پیش بینی شده}})^2$$

$$Param_{new} = Param_{old} - r * \sum {}^{\partial SSE}/_{\partial Param}$$

```
SGD = Stochastic Gradient Descent
optimizer = optim.SGD(model.parameters(), lr=0.01, momentum=0.9)
```

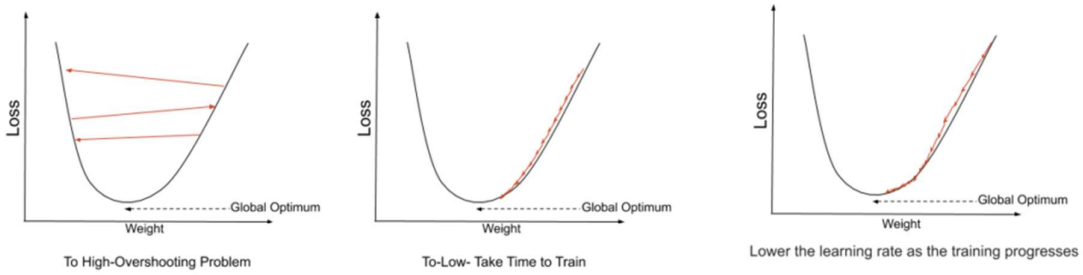
برآورد لحظه سازگار Adaptive Moment Estimation

Adam یک بهینه ساز است که وزن شبکه را ارتقا می دهد. مخفف آن از برآورد لحظه سازگار می آید. Adam به دلیل کارآمد کردن مدل در محاسبات و پیاده سازی استفاده می شود. پارامترهای فوق العاده آن به تنظیم کمی نیاز دارند و طراحی را تسریع می کنند. در واقع Adam جایگزین Stochastic Gradient Descent (SGD) است که یک بهینه ساز سنتی است.

```
Adam Optimizer
optimizer = optim.Adam([var1, var2], lr=0.0001)
```

تفاوت اساسی بین Adam و SGD این است که Adam برای داده های شیب دار و پر نویز مناسب است و Adam میزان یادگیری خود را در طول زمان آموزش تغییر می دهد. Adam ترکیبی از الگوریتم گرادیان تطبیقی (Adagrad) و انتشار میانگین مربع ریشه (RMSProp) است. Adam برای مجموعه داده های ثابت و بزرگ که به طور دلخواه تغییر می کنند مناسب است. [۶]

در خط بیست و دومم از کتابخانه کراس اپتیم، لرنینگ ریت را به نوت بوک معرفی می کنیم. نرخ یادگیری در شبکه های عصبی به تنظیم پارامتر در الگوریتم بهینه سازی اطلاق می شود که در حقیقت اندازه قدم هایی که در هر دوره باید برداشته شود تا چنانچه در بالا ذکر شد بتوان به مینیمم خطا رسید را در بر می گیرد. در شکل ۸-۳ اولین شکل از سمت راست قدم ها مناسب برداشته می شوند و این نرخ متناسب است با پروسه آموزش مدل در شبکه عصبی، در حالیکه در تصویر وسط این نرخ قدم های آموزش بسیار کوچک می باشد در نتیجه زمان رسیدن به نقطه مینیمم بسیار بطول خواهد انجامید. در شکل ۸-۳ اولین تصویر از سمت چپ نرخ قدم های آموزش بسیار بزرگ می باشد بطوریکه ممکن است نقطه اپتیمم در این پرشس های بزرگ از دست برود و نادیده نگاشته شود.



شکل (۸-۳) اندازه قدم های learning rate

در خط هفدهم از کتابخانه کراس مدل، زیر کتابخانه های ، مدل را به نوت بوک معرفی می کنیم. این ساب توابع برای ساماندهی و پردازش داده ها بکار می روند، که می توانند چندین نمونه را در حالتهایی موازی لود نموده و پردازش نمایند.

```
3-import numpy as np
```

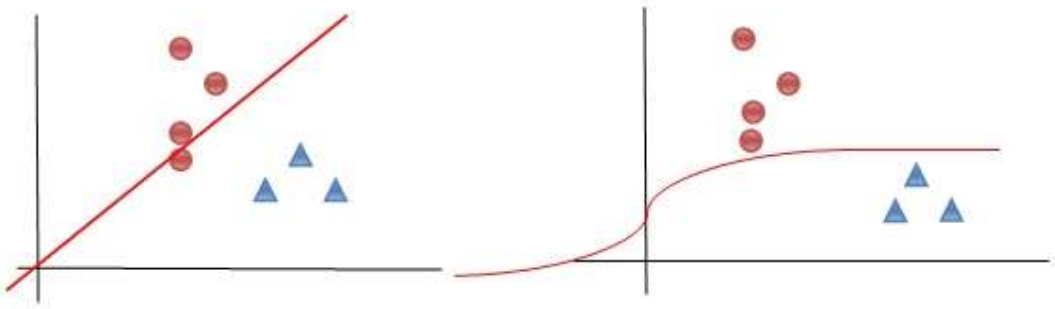
در خط سوم کتابخانه نامپی را به نوت بوک معرفی می کنیم. Numpy یک کتابخانه برای زبان برنامه نویسی پایتون است که پشتیبانی از آرایه ها و ماتریس های بزرگ و چند بعدی را به همراه مجموعه بزرگی از توابع ریاضی سطح بالا برای کار بر روی این آرایه ها اضافه می کند. همچنین دارای توابع برای کار در حوزه جبر خطی ، تبدیل فوریه و ماتریس ها است. NumPy در سال 2005 توسط تراویس اولیفانت ایجاد شد. این یک پروژه اپن سورس است و می توان آزادانه از آن استفاده کرد. NumPy مخفف Numerical Python است.

```
4-import matplotlib.pyplot as plt
```

در خط چهارم pyplot را به نوت بوک معرفی می کنیم.matplotlib pyplot مجموعه ای از توابع است که باعث می شود matplotlib مانند MATLAB کار کند. هر تابع pyplot تغییراتی در یک شکل ایجاد می کند: به عنوان مثال ، یک شکل ایجاد می کند ، یک ناحیه رسم در یک شکل ایجاد می کند ، برخی خطوط را در یک منطقه ترسیم می کند ، طرح را با برچسب ها تزئین می کند و غیره .

```
15-from keras.layers import Input, Flatten, Dropout, Activation
```

در خط پانزدهم تابع فعالسازی را به نوت بوک معرفی می کنیم:



شکل (۹-۳) تابع فعالسازی

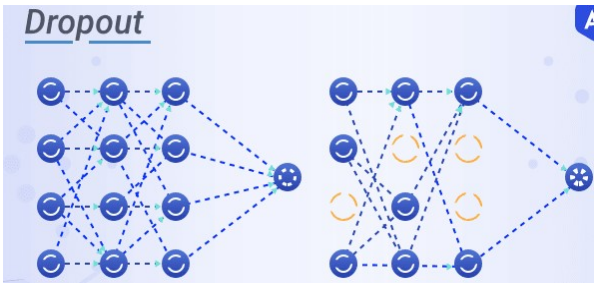
rectified linear=RelU در واقع یک تابع فعالسازی همانند تابع خطی قطعه ای است که در صورت مثبت بودن ورودی، خروجی تغییر نمی کند و در غیر اینصورت در مضرب خاصی ضرب خواهد شد.

$$f(x) = \begin{cases} x & if\ x > 0 \\ ax & otherwise \end{cases}$$

تابع فعال سازی خطی اصلاح شده بر مشکل اضمحلال گرادیان فایق می آید و به مدل ها اجازه می دهد سریعتر یاد بگیرند و عملکرد بهتری داشته باشند.

Dropout رقیق کردن یک روش منظم برای کاهش بیش از حد اتصالات در شبکه های عصبی مصنوعی، بوسیله جلوگیری از سازگاری پیچیده در داده های آموزشی است. این یک روش کارآمد برای انجام میانگین

مدل با شبکه های عصبی است. واژه رقیق سازی به نازک شدن وزن ها اشاره دارد.



شکل (۳-۱۰) تابع فعالسازیDropout

▾ Loading the model

```
1 # loading json and creating model
2 from keras.models import model_from_json
3 json_file = open('model.json', 'r')
4 loaded_model_json = json_file.read()
5 json_file.close()
6 loaded_model = model_from_json(loaded_model_json)
7 # load weights into new model
8 loaded_model.load_weights("saved_models/Emotion_Voice_Detection_Model.h5")
9 print("Loaded model from disk")
10
11 # evaluate loaded model on test data
12 loaded_model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
13 score = loaded_model.evaluate(x_testcnn, y_test, verbose=0)
14 print("%s: %.2f%%" % (loaded_model.metrics_names[1], score[1]*100))
```

```
Loaded model from disk
acc: 72.73%
```

شکل (۳-۱۱) آموزش مدل

در خط ۸ ام شکل ۳-۱۱ مدل رو به جلو حرکت می کند و ورودیها و وزنها در مدل پردازش می شوند. و در خط ۱۲ ام تابع کامپایل اجرا شده و بهینه می شودو در خط ۱۳ ام که همان افتراق بین خروجی واقعی و خروجی پیش بینی شده در انتشار به جلو می باشد، توسط تابع Evaluate محاسبه می گردد. سپس در خط ۱۴ ام درصد نمره اندازه گیری می شود.

فصل ۴:

نتایج و تفسیر آن‌ها

عنوان پایان‌نامه/ تشخیص احساسات گفتاری با استفاده از شبکه عصبی عمیق کانولوشنال

۴-۱- مقدمه

درک انسان از اینکه چه اطلاعاتی در سطح سیگنال گفتاری مفیدتر است (برای تشخیص احساسات) تنها با پیشرفت هایی که دانشمندان در یادگیری ماشین و پردازش سیگنال انجام دادند، بهبود می یابد. در سال‌های اخیر، با محبوبیت یادگیری عمیق در زمینه‌هایی مانند بینایی رایانه یا تشخیص گفتار، تشخیص احساسات نیز به این رویکرد جدید مبتنی بر فناوری یادگیری عمیق متمرکز شد. ممکن است سوال شود که چرا آگاهی عاطفی توسط ماشین ها مهم تلقی می شود.

اولاً برای ایجاد تجربه بهتر به عنوان مثال در بسیاری از سیستم های یادگیری به کمک رایانه، می توان ارائه مطالب یا سرعت یادگیری را با شناخت وضعیت عاطفی یادگیرنده تنظیم کرد و بهترین نتایج را برای آنها به دست آورد. آگاهی عاطفی توسط ماشین‌ها همچنین می‌تواند برای ارائه ابزارهایی به انسان‌ها برای موثرتر کردن آنها استفاده شود، به عنوان مثال در صنعت بازی، می‌توان واکنش گیمر را دید یا شنید و با دانستن نقطه ناامیدی که در طراحی بازی وجود دارد، طراحی بازی را بهبود بخشید.

در بازاریابی تجاری، می‌توان از تشخیص احساسات برای سنجش واکنش بینندگان به مواد بازاریابی استفاده کرد و بر این اساس، مواد را برای دستیابی به اثر مطلوب تنظیم کرد. بنابراین، در همه این کاربردها، نیاز به ساخت ماشین هایی داریم که قادر به درک حالات عاطفی انسان باشند. مردم احساسات را از گفتار، حالات چهره، زبان بدن و غیره درک می کنند و گفتار طبیعی ترین و سریع ترین راه است.

آن فرآیندی که تا یه اینجا انجام شد، بطور خلاصه شامل موارد ذیل می باشد:

۴-۲- پیش پردازش

مرحله 1: نمونه صوتی به عنوان ورودی ارائه می شود.

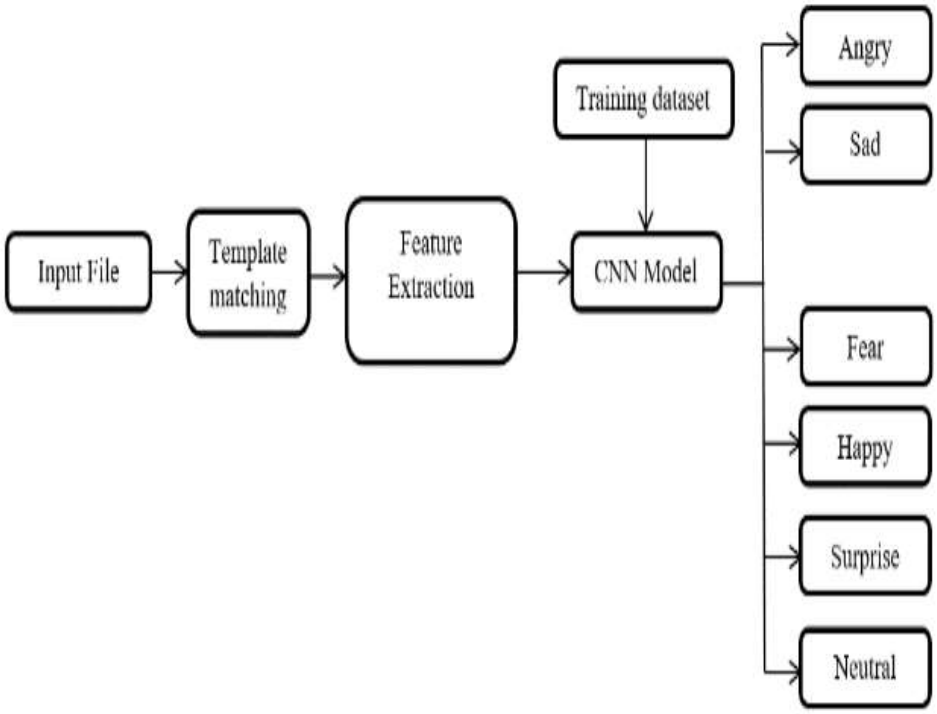
مرحله 2: طیف و شکل موج از فایل صوتی رسم می شود.

مرحله 3: با استفاده از LIBROSA، یک کتابخانه پایتون، معمولاً MFCC (ضریب سپسترال فرکانس مل) را استخراج می کنیم.حدود 20-10.

۴-۳- پردازش

مرحله 4: اختلاط مجدد داده ها، تقسیم آنها در توالی و آزمایش و سپس پس از ساخت یک مدل CNN و موارد زیر لایه ها مانند Drop out, Max Pooling برای آموزش مجموعه داده .

مرحله 5: پیش بینی احساسات صدای انسان از روی آن داده های آموزش دیده (شماره نمونه – ارزش پیش بینی شده – ارزش واقعی)



شکل (۱-۴) مدل

۴-۴- ارائهٔ نتایج

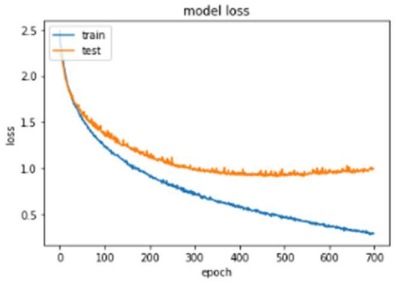
شکل زیر ۲-۴ ، داده های آموزشی train به رنگ آبی، و داده های تستی test به رنگ نارنجی نمایش داده شده است. کاهش خط آموزشی و تستی در مجموعه داده را به وضوح نشان می دهد. همانطور که نمودار می گوید که هر دو خطای "آموزش و آزمایش" با افزایش تعداد دوره های مدل آموزشی کاهش می یابد. خط افقی تعداد دوره های مدل آموزشی و یا همان Epoch ها هستند.

از نمودار شکل ۲-۴ همچنین می توانیم استنباط کنیم که تعداد دوره های مناسب حدود 200 است زیرا دقت داده های آزمون پس از 200 دوره ثابت می ماند. آموزش پس از مدل، ما باید احساسات داده های آزمون را با میانگین 75 درصد به تصویر بکشیم. دقت و حداکثر دقت 82.08 درصد. جدول زیر تصویر ما را با مقادیر واقعی و مقادیر پیش بینی شده نشان می دهد.

Removed the whole training part for avoiding unnecessary long epochs list

```
[ ]  1 cnnhistory=model.fit(x_traincnn, y_train, batch_size=16, epochs=700, validation_data=(x_testcnn, y_test))
```

```
[ ]  1 plt.plot(cnnhistory.history['loss'])
     2 plt.plot(cnnhistory.history['val_loss'])
     3 plt.title('model loss')
     4 plt.ylabel('loss')
     5 plt.xlabel('epoch')
     6 plt.legend(['train', 'test'], loc='upper left')
     7 plt.show()
```



شکل (۲-۴) نتیجه گیری

Actual v/s Predicted emotions

```
1 finaldf[170:180]
```

	actualvalues	predictedvalues
170	female_fearful	female_fearful
171	male_angry	male_angry
172	male_fearful	male_fearful
173	male_happy	male_happy
174	female_happy	female_happy
175	female_angry	female_angry
176	female_angry	female_sad
177	male_sad	male_calm
178	male_angry	male_calm
179	male_sad	male_sad

شکل (۳-۴) نتیجه گیری

شکل ۳-۴ خروجی پیش بینی شده برای ۸ فایل صوتی را نشان می دهد. Actual Values یعنی مقادیر واقعی نشاندهنده برچسب اصلی فایلها می باشند و در سمت راست Predicted Values نشاندهنده خروجی پیش بینی شده توسط مدل کانولوشنال می باشد.

فصل ۵:

نتیجه‌گیری و پیشنهادها

۵-۱- مقدمه

یادگیری عمیق یا Deep Learning زیر مجموعه ای از یادگیری ماشین می باشد و دارای الگوریتم هایی است که از عملکرد و ساختار مغز و شبکه های عصبی که بصورت مصنوعی می باشند ایده گرفته اند از یادگیری عمیق جاهایی استفاده می شود که انسان فعالیت می کند و با استفاده از آن می توان برای حل مسائل و انجام کار ها جایگزین افراد شده و بدون نیاز به نیروی انسانی آن را حل و انجام نمود. کراس یک کتابخانه ی اوپن سورس می باشد که بعنوان فریم ورک توسعه یافته سطح بالا استفاده می شود.

۵-۲- نتایج حاصل از شبیه سازی

۱- به دقت 71 درصد از مدل موجود قبلی رسیده است.
۲- مدل ما با داده های بیشتر بهتر عمل می کند.
۳- مدل هنگام تشخیص صدای مردانه و زنانه بسیار خوب عمل می کند.
٤- مدل با تعداد دفعات بیشتر دوره آموزشی به بهترین و بالاترین دقت می رسد.

۵-۳- نتیجه گیری

پس از ساخت مدل‌های مختلف، مدل CNN بهتری را برای کار تمایز احساسات دریافت کردیم. ما به دقت 71 درصد از مدل موجود قبلی رسیدیم. مدل ما با داده های بیشتر بهتر عمل می کرد. همچنین مدل ما هنگام تشخیص صدای مردانه و زنانه بسیار خوب عمل کرد. پروژه ما را می توان برای ادغام با ربات گسترش داد تا به آن کمک کند تا درک بهتری از حال و هوای انسان مربوطه داشته باشد، که به او کمک می کند مکالمه بهتری داشته باشد و همچنین می تواند با برنامه های مختلف موسیقی برای توصیه آهنگ ها ادغام شود. همچنین می تواند در برنامه های مختلف خرید آنلاین مانند آمازون برای بهبود توصیه محصول به کاربران استفاده شود. علاوه بر این، در سال‌های آینده می‌توانیم یک مدل توالی به دنباله بسازیم تا صدایی با احساسات متفاوت ایجاد کنیم.

مراجع و منابع

[1] A. Deshpande, (2016), A Beginner's Guide To Understanding Convolutional Neural Networks, GitHub repository, https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/.

[2] B. N. Kaushik, (2020), A Hybrid Technique using CNN+LSTM for Speech Emotion Recognition, International Journal of Engineering and Advanced Technology 9(5):1126-1130.

[3] H. Murugan, (2020), Speech Emotion Recognition Using CNN, International Journal of Psychosocial Rehabilitation 24(8), DOI: 10.37200/IJPR/V24I8/PR280260.

[4] J. SidorovaToni, B. Badia, (2008), ESEDA: A Tool for Enhanced Speech Emotion Detection and Analysis, Conference: Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS '08. International Conference on, DOI: 10.1109/AXMEDIS.2008.39.

[5] Y. Nam, C. Lee, (2021), Cascaded Convolutional Neural Network Architecture for Speech Emotion Recognition in Noisy Conditions. https:// doi.org/10.3390/s21134399.

[6] M. R. Izadi, Y. Fang, R. Stevenson and L. Lin, (2020), Optimization of Graph Neural Networks with Natural Gradient Descent, IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 171-179, doi: 10.1109/BigData50022.2020.9378063.

پیوست‌ها

پیوست الف

```
model = Sequential()

model.add(Conv1D(256, 5,padding='same',
                 input_shape=(216,1)))
model.add(Activation('relu'))
model.add(Conv1D(128, 5,padding='same'))
model.add(Activation('relu'))
model.add(Dropout(0.1))
model.add(MaxPooling1D(pool_size=(8)))
model.add(Conv1D(128, 5,padding='same',))
model.add(Activation('relu'))
#model.add(Conv1D(128, 5,padding='same',))
#model.add(Activation('relu'))
#model.add(Conv1D(128, 5,padding='same',))
#model.add(Activation('relu'))
#model.add(Dropout(0.2))
model.add(Conv1D(128, 5,padding='same',))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(10))
model.add(Activation('softmax'))
opt = keras.optimizers.rmsprop(lr=0.00001, decay=1e-6)


model_name = 'Emotion_Voice_Detection_Model.h5'
save_dir = os.path.join(os.getcwd(), 'saved_models')
# Save model and weights
if not os.path.isdir(save_dir):
    os.makedirs(save_dir)
model_path = os.path.join(save_dir, model_name)
model.save(model_path)
print('Saved trained model at %s ' % model_path)


# loading json and creating model
from keras.models import model_from_json
json_file = open('model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# load weights into new model
loaded_model.load_weights("saved_models/Emotion_Voice_Detection_Model.h
5")
print("Loaded model from disk")

# evaluate loaded model on test data
```

```
loaded_model.compile(loss='categorical_crossentropy', optimizer=opt, me
trics=['accuracy'])
score = loaded_model.evaluate(x_testcnn, y_test, verbose=0)
print("%s: %.2f%%" % (loaded_model.metrics_names[1], score[1]*100))
```

Abstract:

The main goal of this research is based on Emotional Speech Detection, which is related to the different actors as the inputs. In this thesis, there is Convolutional Neural Network. The most important things here is to compute how much CNN can improve the acceleration and accuracy for model learning instead of using another algorithm. The purpose is to predict the growth of velocity parameter while holing on precision parameters. In this research, the mentioned parameters are extracted from Google Colab.

Keywords: Speech emotion, Deep learning, Keras, Tensor flow, CNN

BSc/ MSc/ PhD Thesis Title

Emotional Speech Detection By Convolutional Neural Network

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Bachelor of Science / Master of Science / Doctor of Philosophy in - engineering - Orientation

By:

Supervisor:

Dr.

Advisor:

Dr.

October 2021