دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده برق

پایان‌نامه کارشناسی

گرایش کنترل

عنوان پایان نامه – شناسایی احساسات با استفاده از یادگیری عمیق

نگارش

رامتین عسگریان امیری

استاد راهنما

فرزانه عبدالهی

مرداد ۱۴۰۰

صفحه فرم ارزیابی و تصویب پایان نامه- فرم تأیید اعضاء کمیته دفاع

به نام خدا

تاریخ:

تعهدنامه اصالت اثر

اینجانب رامتین عسگریان امیری متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

رامتین عسگریان امیری

امضا

تشکر

باسپاس فراوان از استاد محترم،دکتر فرزانه عبداللهی ، که بنده را در انجام این پروژه همراهـی و راهنمایی کردند.

چکیده

درک احساس شتری برای مشاغل و کسب و کار های مختلف با استفاده از فناوری های به روز و نوین فراهم شده است. این فناوری توانایی فروش در لحظه مناسب را با توجه به تحلیل احساس مصرف-کننده دارد. هوش ایموشن، دانشی رو به رشد محسوب است که نه تنها در تبلغات، بلکـه در اسـتارت‌آپ های جدید، مراقبت های بهداشتی، ابزار های دیجیتالی پوشیدنی،ارتباط انسان وربـات، آمـوزش و مـوارد دیگر می‌تواند تاثیرات زیادی به همراه داشته باشد.

افزایش اهمیت در دقت و صحت این امر را به سمت استفاده از سیستم های تشـخیص احسـاس چهره به کمک شبکه‌های عصبی سوق داده است. در این پروژه سعی شده تا با نگاهی به مطالعات پیشین در زمینه‌های مشابه، به بررسی توانایی های یادگیری عمیق در راستای دسته بندی حالات چهره و بهبود عملکرد طبقه‌بندی‌کننده‌ها پرداخته شود. در ایـن راسـتا از دیتاسـت رایـج و در دسـترس FER-2013 استفاده شده‌است.

در این رساله با الهام گیری از شبکه‌ی عصبی خاص، سعی بر دست‌یابی بـه شـبکه‌ی عصـبی بـا هدف شناسایی احساسات است. در انتها شبکه‌ای با دقت ۷۱ درصد حاصل شده‌است.

واژه‌های کلیدی:

احساسات، تشخیص چهره، یادگیری عمیق، ساخت شبکه عصبی کانولوشن

فهرست مطالب

صفحه

تشکر **۱**

چکیده ..... **أ**

فصل اول مقدمه **۱**

۱-۱- مقدمه ۲

۱-۲- مروری بر پژوهشهای پیشین ۴

فصل دوم یادگیری ماشین و شبکه‌های پیچشی **۸**

۲-۱- یادگیری عمیق ۹

۲-۲- دسته بندی روشهای یادگیری عمیق ۱۰

۲-۲-۱- یادگیری با نظارت ۱۱

۲-۲-۲- یادگیری بدون نظارت ۱۱

۲-۲-۳- یادگیری نیمه نظارتی ۱۲

۲-۲-۴- یادگیری تقویتی ۱۲

۲-۳- شبکه عصبی پیچشی (CNN) ۱۳

۲-۳-۱- معماری شبکههای عصبی پیچشی ۱۵

۲-۳-۱-۱-لایه کانولوشن ۱۶

۲-۳-۱-۲-لایه پولینگ ۱۸

۲-۳-۱-۳-لایه های کاملا متصل ۲۰

۲-۳-۲- توابع فعالساز ۲۰

۲-۳-۲-۱-تابع فعالساز سیگموید ۲۱

۲-۳-۲-۲-تابع فعالساز تانژانت ۲۲

۲-۳-۲-۳-تابع فعالساز رلو (ReLU) ۲۳

۲-۴- کاربردهای شبکه‌های عصبی کانولوشنی ۲۴

۲-۴-۱- تشخیص تصاویر ۲۴

۲-۴-۲- تشخیص ویدیو ۲۵

۲-۴-۳- پردازش زبان طبیعی ۲۵

فصل سوم شناسایی احساسات و چهره **۲۶**

۳-۱- شبکه عصبی و احساسات ۲۷

۳-۱-۱- انواع احساسات ۲۸

۳-۱-۲-کاربردهای فناوری تشخیص احساسات ۲۹

۳-۲- دیتاست های تشخیص احساسات ۲۹

۳-۲-۱- معرفی تعدادی از دیتاست های مرتبط ۳۰

۳-۳- دیتاست FER ۳۲
۳-۳-۱- بازنگری تصاویر ۳۳
۳-۳-۲- بهینه‌سازی دیتاست ۳۴
۳-۴- چارچوب تشخیص اشیا ویولا-جونز ۳۴
۳-۴-۱- انواع ویژگی ها و ارزیابی ۳۵
۳-۴-۲- مستطیل‌های ویژگی ۳۶
۳-۴-۳- ویژگی‌های هار ۳۷
۳-۴-۳-۱-استفاده ازآدابوست ۳۸
۳-۴-۳-۲-کلاس‌بندی آبشاری ۳۹
فصل چهارم معماری شبکه‌های عصبی ۴۱
۴-۱- انواع شبکه‌های عصبی ۴۲
۴-۱-۱- 5-لنت ۴۳
۴-۱-۲- الکسنت ۴۴
۴-۱-۳- VGG-16 ..... ۴۵
۴-۱-۴- اینسپشن-۱(گوگل‌نت) ۴۶
۴-۱-۵-اینسپشن-۳ ..... ۴۸
۴-۱-۶- رس‌نت ۵۰
۴-۱-۷-اکسپشن ۵۲
۴-۱-۷-۱- مقایسه در دقت ۵۳
۴-۱-۸-اینسپشن-رس‌نت ۵۴
فصل پنجم تعاریف در پیاده‌سازی و شبیه‌سازی ۵۶
۵-۱- بیان مسئله ۵۷
۵-۲- پردازش تصویر ۵۷
۵-۲-۱- تصویر خاکستری در پردازش ۵۸
۵-۲-۲- آستانه‌گذاری در تصاویر ۵۹
۵-۲-۳- آماده سازی دیتاست ۶۰
۵-۲-۴- دسته بندی تصاویر برای آموزش و تست ۶۰
۵-۳- بهینه‌سازی ۶۱
۵-۳-۱- بهینه‌ساز نزولی گرادیان ۶۲
۵-۳-۱-۱- نزول گرادیان دسته‌ای ۶۲
۵-۳-۱-۲- نزول گرادیان تصادفی ۶۳
۵-۳-۱-۳- نزول گرادیان دسته‌ای کوچک ۶۴
۵-۳-۲- الگوریتمهای تطبیقی ۶۴
۵-۳-۲-۱-آداگارد ۶۴
۵-۳-۲-۲-آدادلتا ۶۶

۵-۳-۲-۳-آدام ۶۶
۵-۴- اصطلاحات یادگیری ماشین ۶۷
۵-۴-۱- ایپچ ۶۷
۵-۴-۲- اندازه دسته ۶۸
۵-۴-۳- تکرار ۶۹
۵-۵- معیارهای ارزیابی شبکه ۶۹
۵-۵-۱- دسته‌بندی نتایج ۷۰
۵-۵-۲- روش‌های ارزیابی الگوریتم ۷۰
۵-۵-۲-۱- ماتریس درهم‌ریختگی ۷۰
۵-۵-۲-۲- دقت ۷۱
۵-۵-۲-۳- صحت ۷۱
۵-۵-۲-۴- فراخوانی ۷۲
۵-۵-۲-۵- F-measure/F1 Score ۷۲
۵-۵-۲-۶- خاصیت ۷۳
۵-۵-۲-۷- MCC ۷۳
۵-۵-۳- نرخ nام خطا ۷۴
۵-۵-۴- منحنی ROC ۷۵
۵-۵-۵- AUC ۷۶
۵-۶- پیاده‌سازی ۷۶
۵-۶-۱- افزایش پارامترها ۸۰
۵-۶-۲- بهبودسازی مدل پیچیده ۸۲
فصل ششم جمع‌بندی و نتیجه‌گیری و پیشنهادات ۸۵
منابع و مراجع ۸۸
پیوست‌ها ۹۲
Abstract ۱۰۴

فهرست اشکال

صفحه

شکل ۲-۱- نمایی از پیکرسازی ساختمان شبکه‌ی عصبی پیچشی ۱۵
شکل ۲-۲- یک مثال از عمکرد لایه کانولوشن ۱۷
شکل ۲-۳- چگونگی رفتار لایه های متوالی کانولوشن ۱۸
شکل ۲-۴- مثالی از عملکرد دو نوع پولینگ ۱۹
شکل ۲-۵- تابع فعال‌ساز سیگموید ۲۲
شکل ۲-۶- تابع فعال‌ساز تانژانت ۲۲
شکل ۲-۷- تابع فعال‌ساز رلو ۲۴
شکل ۳-۱- تعدادی مثال از عکس های دیتاست FER-2013 ۳۳
شکل ۳-۲- مثال‌هایی از ویژگی‌های مستطیلی ۳۶
شکل ۳-۳- جایگیری ویژکی هار ۳۸
شکل ۴-۱- عملکرد تعدادی از معماری‌های شبکه عصبی در پروژه ایمیج‌نت ۴۳
شکل ۴-۲- معماری شبکه عصبی لنت-۵ ۴۴
شکل ۴-۳- معماری شبکه‌ی عصبی الکس‌نت ۴۵
شکل ۴-۴- معماری شبکه‌ی عصبی VGG-16 ۴۶
شکل ۴-۵- معماری شبکه‌ی عصبی اینسپشن-۱ ۴۸
شکل ۴-۶- معماری شبکه‌ی عصبی نسخه سوم اینسپشن ۴۹
شکل ۴-۷- معماری زیر ساخت‌های نسخه سوم اینسپشن ۵۰
شکل ۴-۸- معماری شبکه‌ی عصبی رس‌نت ۵۲
شکل ۴-۹- معماری شبکه‌ی عصبی اکسپشن ۵۳
شکل ۴-۱۰- مقایسه دقت دو معماری اکسپشن و اینسپشن-۳ ۵۴
شکل ۴-۱۱- معماری شبکه‌ی عصبی اینسپشن-رس‌نت ۵۵
شکل ۵-۱- تصویر رنگی و سه صفحه قرمز ۵۸
شکل ۵-۲- معیارهای ارزیابی شبکه‌ی عصبی ۷۴
شکل ۵-۳- مدل الهام گرفته‌شده از اکسپشن ۷۷
شکل ۵-۴- نمودار دقت برای مدل اصلی ۷۸
شکل ۵-۵- نمودار اعتبارسنچی زیان برای مدل اصلی ۷۹
شکل ۵-۶- ماتریس درهم‌ریختگی نرمال شده ۸۰
شکل ۵-۷- نمودار دقت برای مدل افزایش پارامتر داده‌شده ۸۱
شکل ۵-۸- نمودار اعتبارسنجی تابع زیان برای مدل افزایش پارامتر داده‌شده ۸۱
شکل ۵-۹- نمودار دقت برای بهبودیافته‌ی مدل افزایش پارامتر داده‌شده ۸۳

شکل ۵-۱۰- نمودار اعتبارسنجی تابع زیان برای بهبودیافته‌ی مدل افزایش پارامتر داده‌شده..... ۸۳

فهرست جداول

صفحه

جدول۱-۱ تعدادی از دیتاست‌های تشخیص احساسات..... ۲۹
جدول۱-۲ تقسیم‌بندی داده‌ها در FER..... ۳۰

فصل اول

مقدمه

۱-۱- مقدمه

احساسات نقش مهمی و موثر در روابط انسان با انسـان ایفـا می‌کننـد. در واقـع، در بسـیاری از موقعیت‌ها، هوش عاطفی نسبت به IQ[1] برای تعامل موفق انسان، مهم‌تر است. همچنـین شـواهد مهمـی حاکی از آن است که، یادگیری عقلانی در انسان‌ها به احساسات وابسته است. از این رو, سنجش عـاطفی و آنالیز احساسی, کلید پیشـرفت در هـوش مصـنوعی و تمـامی زمینـه‌های تحقیقـاتی کـه از آن ریشـه می‌گیرند هسـتند.[۱] بـه علـاوه, آن‌هـا کاربردهـایی در چنـدین سـناریوی مختلـف و تعـداد زیـادی از شرکت‌های بزرگ و کوچک وجود دارد، که تجزیـه و تحلیـل احساسـات بـه عنـوان بخشـی ازخواسـته و ماموریت آن‌ها است. شناسایی احساسات را می‌توان برای ایجاد و ساخت تراکم نظر و بازبینی وب‌سایت‌ها مورد استفاده قرار داد. این ابزار فقط به نقدهای محصولات محدود نمی‌شوند بلکه موضوعات گسترده‌تر از قبیل مسائل سیاسی و معضل های اجتماعی نیز می تواند در بر بگیرد.

درک عاطفی و آنالیز احساس نیز پتانسیل بالایی به عنوان یک فـن‌آوری بـرای بخـش فرعـی در سیستم‌های دیگر دارند. آن‌ها می‌توانند قابلیت‌های مدیریت ارتباط بـا مشـتری و سیسـتم‌های توصـیه و معرفی کالا به مشتری را افزایش دهند. این افزایش قابلیت برای مثال به آن‌ها اجازه‌ی مشخص کردن این که، کدام گروه از مشتریان به طور خاص از کالا ی مدنظر راضی هستند می‌دهد.[۲]

علاوه بر این , الگوریتم های یادگیری ماشین نقـش قابـل‌توجهی در تشـخیص الگـو و مشـکلات طبقه‌بندی الگو، به ویژه در حالات چهره واحساسات در چهره طی چند دهه گذشته بازی کرده‌اند.[۳-۵]

در دهه‌های‌گذشته قابلیت‌های " الگوریتمهای بر پایه‌ی یادگیری عمیق "رشد چشم‌گیری داشته است." یادگیری عمیق " یک زیرمجموعه از هوش مصنوعی است که در آن یک الگوریتم کامپیوتری بـه تحلیل داده های خام پرداخته و ویژگی‌های لازم برای شناسایی الگوهـای ویژگـی در داده‌هـا را بـه طـور خودکار یاد می‌گیرد. استفاده از " الگوریتم‌های یادگیری عمیق "در شناسایی اجسام و طبقه‌بندی تصاویر نیز با پیشرفت‌های بسیاری همراه بوده است که این پیشرفت‌ها در زمینه بازرگانی و تبلیغاتی نیز تحولاتی را به وجود آورده است.[۶-۸]

[1] intelligence quotient

هدف اصلی این پژوهش، تشخیص هفت حالت احساسی افراد با استفاده از ویژگی‌های تصویر یک صورت است.این هفت احساس برپایه‌ی نظریه‌های تفکیک‌سازی احساسات از عوامل جدا از هم است. هر انسان با تحلیل چهره، صدا و زبان بدن به نتیجه‌ای برای طبقه‌بندی احساسات می رسد.[۱] به علاوه دقت انسان برای طبقه‌بندی تصویری از چهره در یکی از ۷ احساسات مختلف، ٪۶۵± ٪۵ است.[۹] برای حداقل کردن زمان جواب دهی و بهینه کردن فرآیند، شبکه‌ی عصبی پیشنهاد شده‌است. در شبکه‌های عصبی , تشخیص چهره در دو مرحله با استفاده از پیکسل‌های تصویر اجرا می‌شود.[۱۰و۱۱]

همانطور که بیان شد، مرحله‌ی اول تشخیص چهره با استفاده از چهارچوب شناسایی بر پایه‌ی هار است سپس در مرحله بعد شبکه‌ی عصبی پیشنهاد شده سعی در شناسایی احساس دارد.

۱-۲- مروری بر پژوهشهای پیشین

در سال‌های اخیر، حوزه هوش مصنوعی پیشرفت سریعی داشته‌است. در عصر دیجیتال امروز، نیاز مبرمی به HCI[۲] هوشمند وجود دارد. به عنوان یک شاخه مهم از تحقیقات هوش مصنوعی، شناسایی احساسات به طور فزاینده‌ای، توجه شدید پژوهشگران را به خود جلب کرده‌است. در حال حاضر تحقیقات در مورد تشخیص احساسات بر روی موضوعات و مساله‌های زیر متمرکز است:

۱. همبستگی بین انواع مختلف سیگنال‌های فیزیولوژیکی و احساسات
۲. روش‌های انتخاب محرک‌ها برای القای حالات احساسی مورد انتظار
۳. الگوریتم‌های استخراج مشخصه های احساسات
۴. روش‌های بازشناسی احساسات مبتنی بر ترکیب اطلاعات چند وجهی و چند جانبه

در ادامه، مرور کوتاهی از تحولات تحقیقات پیشین و فعلی در این جهت‌ها ارائه خواهیم داد.

شناسایی خودکار چهره در دهه ۱۹۶۰ اولین قدم ها را سپری کرده بود. وودی آلن، هلن چان و چارلز بیسون با استفاده از کامپیوتر برای تشخیص چهره‌های انسانی کار کردند. پروژه تشخیص چهره آن‌ها "Man_Machine " نامیده شد، زیرا مختصات ویژگی‌های صورت در یک عکس باید توسط یک انسان قبل از اینکه بتوانند توسط کامپیوتر برای شناسایی استفاده شوند، مشخص می‌شد. در یک لوح گرافیکی، یک انسان باید مختصات ویژگی‌های چهره مانند مراکز چشم‌ها، داخل و خارج چشم‌ها و پیک رویش در خط موی سر و ابرو را مشخص کند. این مختصات برای محاسبه ۲۰ فاصله از جمله عرض دهان و چشم‌ها استفاده می‌شد. یک انسان می‌توانست حدود ۴۰ عکس را در یک ساعت با این روش پردازش کند و یک مجموعه داده از فواصل محاسبه‌شده ایجاد کند. سپس یک کامپیوتر به طور اتوماتیک فواصل را برای هر عکس مقایسه می‌کند, تفاوت بین فواصل را محاسبه کرده و حدسی را به عنوان یک همسان ممکن ارائه می‌کند.[۱۲]

[2] Human-computer interaction

در ۱۹۷۰, تکیو کند[3] یک سیستم تطبیق چهره را معرفی کردکه ویژگی‌های آناتومی مانند چانه را بررسی می‌کرد و نسبت فاصله بین ویژگی‌های چهره را، بدون دخالت انسان محاسبه می‌کرد. آزمایش‌های بعدی نشان داد که این سیستم نمی‌تواند همیشه ویژگی‌های صورت را تایید کند. با این وجود, علاقه به این موضوع افزایش یافت و در سال ۱۹۷۷ توکیو کند[5] اولین کتاب مفصل را در مورد تکنولوژی تشخیص چهره با ارائه تمام تحقیقات و پژوهش‌های پیشین را منتشر کرد.[۱۳]

در سال ۱۹۹۳, سازمان پروژه تحقیقاتی پیشرفته دفاع (DARPA) و آزمایشگاه تحقیقات ارتش (ARL) برنامه فن‌آوری تشخیص چهره را برای توسعه قابلیت‌های تشخیص چهره اتوماتیک، راه‌اندازی کردند. این فناوری می‌تواند در محیط زندگی به کار گرفته شود تا به پرسنل امنیتی, اطلاعات و نیروهای انتظامی در اجرای وظایف خود کمک کند. سیستم‌های تشخیص چهره که در آزمایشگاه‌های تحقیقاتی مورد ارزیابی قرار گرفتند و تست‌هایFERET نشان دادند که در حالی که روش‌های عملکرد سیستم‌های تشخیص چهره خودکار موجود، متفاوت است، تعداد کمی از روش‌های موجود می‌تواند برای تشخیص چهره‌ها در تصاویر محیط‌های کنترل‌شده، هنوز مورد استفاده قرار گیرد. تست‌های FERET باعث ایجاد سه شرکت آمریکایی شدند که سیستم‌های بازشناسی چهره خودکار را می‌فروختند. شرکت Vision و شرکت Miros در سال ۱۹۹۴ توسط محققانی که از نتایج آزمایش‌ها FERET به عنوان یک فرصت فروش استفاده کردند تاسیس شد.[۱۴]

تا دهه ۱۹۹۰ سیستم‌های تشخیص چهره در ابتدا با استفاده از پرتره‌های تصویری چهره‌های انسانی توسعه داده شدند. در این توسعه جای‌گیری صورت با توجه به المان های اصلی صورت امکان‌پذیر بود(PCA)[4]. روش PCA برای تشخیص چهره نیز به نام "Eigenface" شناخته می‌شود و توسط متیو ترک و آلکس پنتالد ساخته شده‌است. ترک و پنتالد رویکرد مفهومی تئوری، کار هون لوو و آنالیز فاکتوری را ترکیب کردند تا یک مدل خطی را توسعه دهند. Eigenfaces براساس ویژگی‌های جهانی و متعامد در صورت‌های انسانی تعیین می‌شوند. صورت انسان به عنوان یک ترکیب وزن دار از تعدادی از Eigenfaces مشخص می‌شود.[۱۵]

[3] Takeo kanade

[4] Principle Component Analysis

با توجه به پیشرفت های حاصل در شناشایی چهره استفاده از آن در بسترهای گستردهای رایج شد. شناسایی احساسات از نمونههای این گسترش علم است.

یکی از دانشجویان فارغ التحصیل به نام, رنا ال کلیوبی یکی از اولین افرادی بود که شروع به آزمایش این رویکرد کرد. در سال ۲۰۰۱ پس از انتقال از مصر به دانشگاه کمبریج , مدرک دکترای خود را در علوم کامپیوتر به دست آورد و متوجه شد که بیشتر وقتش را با کامپیوتر خود بودهاست تا با دیگر افراد سپری کند. او به این نتیجه رسید که اگر او بتواند به کامپیوتر آموزش دهد تا به حالت عاطفی خود واکنش نشان دهد, زمان تنهایی و دوری از خانواده و دوستان را کمتر احساس خواهد کرد.

کلیوبی بقیه مطالعات دکترای خود را وقف کار بر روی این مشکل کرد و در نهایت نرمافزاری برای کمک به کودکان مبتلا به سندروم آسپرگر ساخت تا به کمک آن کودکان قادر شوند به حالتهای چهره پاسخ دهند. او آن را " کمک به شنوایی عاطفی [۵]" نامید.

آمازون, مایکروسافت و IBM در حال حاضر، تحلیل احساسات را به عنوان یکی از محصولات در حوزهی تشخیص چهره خود تبلیغ میکنند و تعدادی از شرکتهای کوچک مانند کاریوس وآیریس, ارائه خدمات مشابه به افکتیوا را ارائه دادند .فراتر از تحقیقات بازار، فنآوری تشخیص احساسات در حال حاضر برای نظارت و تشخیص اختلال رانندگان، تست تجربه کاربر برای بازیهای ویدیویی و کمک به متخصصان پزشکی برای ارزیابی رفاه بیماران مورد استفاده قرار میگیرد.[۱۶]

[5] emotional hearing aid

فصل دوم

یادگیری ماشین و شبکه‌های پیچشی

۲-۱- یادگیری عمیق

یادگیری عمیق یکی از روش های جدید با رشد سریع در زمینه ی یـادگیری ماشـین اسـت. در این روش تلاش می شود تا استخراج ویژگی ها و اطلاعات مفید از داده هایی با مقیاس بزرگ، با اسـتفاده از شبکه های عصبی عمیق چند لایه ای (DNNها)[۱]، استخراج شود تا بتوان از داده هایی مانند تصـاویر، صداها و متن ها، اطلاعات مفید به دست آورد. یادگیری عمیق به صورت عمومی دارای دو ویژگی است:

۱)لایه های چندگانه از واحد های پردازش غیر خطی

۲) یادگیری با سرپرست یا بدون سرپرست از ویژگی ها در هر لایه

قالب کاری اولیه برای یادگیری عمیق بر اساس شـبکه هـای عصـبی مصـنوعی (ANNهـا)[۲] در دهه‌ی ۱۹۸۰ ایجاد شد، در حالی که تاثیر واقعی این روش های یادگیری عمیـق در سـال۲۰۰۶ نمایـان شد و از آن زمان تا کنون، یادگیری عمیق در گستره‌ی زمینه هـای مختلـف، شـامل شناسـایی خودکـار گفتار، شناسایی تصویر، پردازش طبیعی زبان، شناسایی دارو و انفورماتیک زیسـتی مـورد اسـتفاده قـرار گرفته است.

با تمرکز و توجه بیشتر به این روش و تلاش بیشتر، یادگیری عمیق در سال‌های اخیـر پیشـرفت زیادی داشته است و به صورت گسترده در صنایع مختلف مورد استفاده قرار گرفته است. به عنوان مثال، شبکه های باور عمیق (DBNها)[۳] و ماشین های محدود بلتزمن (RBM)[۴]، در شناسایی تصـویر، صـدا و پردازش طبیعی زبان مورد استفاده قرار گرفته اند. شبکه های عصـبی پیچشـی (CNNهـا)[۵] بـه صـورت گسترده در زمینه ی شناسایی تصویر، بخش بندی تصویر، شناسایی ویدئو و پردازش زبان طبیعـی مـورد

[1] deep neural network

[2] Artificial neural networks

[3] Deep belief networks

[4] Restricted Boltzmann machine

[5] Convolutional neural network

استفاده قرار گرفته اند. شبکه های عصبی بازگشتی (RNNها)[1] هم دسته ای دیگر از شبکه های عصبی هستند که رفتار پویایی از خودشان نشان می دهند و در این شبکه ها، نورون های عصبی با گام های زمانی با هم در ارتباط هستند. این RNNها مهم ترین ابزار برای کار با داده‌های متوالی هستند و در پردازش زبان طبیعی و شناسایی دست خط، به صورت رایج مورد استفاده قرار می گیرند.

یادگیری ماشین[2] امروزه در تجزیه و تحلیل مقادیر عظیمی از انواع داده ها، از جمله داده های پزشکی، تبلیغاتی و هدف‌های شناسایی در ضمینه‌های مختلف از جمله شناسایی احساسات کاربرد دارد. این شناسایی ارائه شده توسط سیستم‌های مذکور، سریع و دقیق است. تلفیق یادگیری ماشین با فناوری های شناختی مانند هوش مصنوعی می تواند به پردازش حجم زیادی از داده ها در کنار صحت بالا بپردازد. در این علم، علومی همچون ریاضیات، آمار، علم کامپیوتر و شاخه هایی از علوم برق کاربرد فراوان دارند. در ادامه به دسته‌بندی یادگیری ماشین می‌پردازیم و مختصرا آن ها را شرح خواهیم داد.[۱۶]

۲-۲- دسته بندی روش‌های یادگیری عمیق[۱۷و۱۸]

علم یادگیری عمیق با توجه به روش‌های پیاده‌سازی استفاده شده در آن و اجزای اصلی تشکیل-دهنده‌ی آن در دسته‌بندی‌های متفاوتی قرار می‌گیرند. این دسته ها به چهار گروه اصلی زیر تقسیم می شوند که عبارتند از:

۱. یادگیری با نظارت
۲. یادگیری بدون نظارت
۳. یادگیری نیمه نظارتی
۴. یادگیری تقویتی

[1] Recurrent neural network

[2] Machine learning

۲-۲-۱- یادگیری با نظارت[1]

الگوریتم های با ناظر یا تحت نظارت از دسته الگوریتم های یادگیری ماشین است که، در این گروه آنچه را که در گذشته آموخته‌شده را در آینده استفاده می‌کند و از آموخته‌های خود برای پیش‌بینی استفاده می‌کند. در روش با نظارت مجموعه با استفاده از داده های آموزش که صحت آن ها تایید شده-است، به ساخت الگوریتم و تابع شناختی که منجر به پیش‌بینی خروجی های داده های ورودی می‌شود، می‌پردازد. این سیستم اگر به اندازه کافی آموزش داده شود، قادر خواهد بود تا برای داده های جدید خروجی مورد نظر را ارائه دهد. این الگوریتم یادگیری نیز قادر است با مقایسه خروجی های تعیین شده برای داده های آموزش و خروجی های به‌دست آمده از الگوریتم، خطا را محاسبه کرده و مدل را متناسب با خطا بهبود ببخشد. در کل سیستم تلاش می‌کند تا تقابلات خود با یک محیط پویا را از طریق آزمون و خطا بهینه نماید.[۱۷]

انواع شبکه های با نظارت: DNN, CNN و RNN

۲-۲-۲- یادگیری بدون نظارت[2]

الگوریتم‌های یادگیری بدون نظارت همان‌گونه که از نام‌گذاری آن مشخص است، در مقابل مفهوم یادگیری‌های با ناظر قرار دارد. در این گروه از یادگیری اطلاعات نادقیق که به معنی دیتاست بدون طبقه بندی و برچسب‌گذاری یا ورودی های بدون خروجی کامل است،استفاده می‌شود. در یادگیری بدون ناظر، سیستم قادر است تا ارتباط و تابع برای توصیف ساختار داده های بدون برچسب که به آن‌ها ارتباط پنهان گفته می‌شود را استنباط کند. خروجی صحیح در این یادگیری حاصل نمی‌شود بلکه ساختار تابعی از کاوش داده‌ها استنباط می‌شود.[۱۷]

نمونه‌هایی ازانواع شبکه‌های بدون نظارت: LOF [3], Hierarchical Clustering و K-means

[1] Supervised learning

[2] Unsupervised learning

[3] Local Outlier Factor

۲-۲-۳- یادگیری نیمه نظارتی

الگوریتم های یادگیری ماشین نیمه نظارت شده، ویژگی‌هایی در بین روش یادگیری با ناظر و بدون ناظر قرار می‌گیرند. زیرا در این روش، بخشی از داده‌های ارائه شده برای آموزش دارای برچسب هستند و برخی بدون برچسب و دسته بندی. به طور معمول در روش یادگیری نیمه نظارت شده مقدار کمی از داده‌های دارای برچسب و مقدار زیادی از داده ها بدون برچسب هستند. سیستم‌هایی که از این روش استفاده می‌کنند، می‌توانند به میزان قابل توجهی دقت یادگیری را افزایش دهند. در این روش‌ها به کامپیوتر تنها یک سیگنال آموزشی ناقص داده می‌شود. منظور از سیگنال آموزشی ناقص، داده‌هایی است که بسیاری از خروجی‌های آن از دسترس خارج هستند.

انواع شبکه‌های عمیق نیمه نظارتی: GAN[1], DRL[2] و RNN

۲-۲-۴- یادگیری تقویتی[3]

روش یادگیری عمیق تقویتی یا تقویت شده به آن دسته از یادگیری ماشین اتلاغ می‌گردد که برای رسیدن به هدفی خاص و از پیش تعیین شده، مثلا برنده شدن در یک بازی کامپیوتری، تلاش می‌کند. در این روش ماشین توسط بهره‌گیری از روش آزمون و خطا برای تقویت مدل خود پیشرفت می‌کند. این خطاها از بازخوردهای مثبت و منفی در رابطه با عملکرد خود محاسبه شده و از همین رو یادگیری تقویتی نام گرفته‌است. در نهایت زمانی که تجربه کافی به دست آورد، به هدف مشخص شده خواهد رسید. به دو روش که یکی بر پایه مقدار دهی[4] و دیگری بر پایه قانون‌گذاری[5] است، انجام می‌گیرد.[۱۸]

[1] Generative Adversarial Network

[2] Deep reinforcement learning

[3] Reinforcement learning

[4] Value-Based

[5] Policy-Based

۲-۳- شبکه عصبی پیچشی (CNN)

در یادگیری عمیق, شبکه عصبی کانولوشنال (convolutional یا ConvNet) دسته‌ای از شبکه عصبی عمیق است که بیشتر برای تجزیه و تحلیل و آنالیز تصاویر بصری به کار می‌رود. آن‌ها همچنین به عنوان شبکه‌های عصبی تغییر ناپذیر یا (SIANN[1]), براساس معماری وزن‌دهی کرنل های کانولوشنی[2] یا فیلتر که در امتداد ورودی جا به جا می‌شوند، عمل می‌کنند تا به تابع معادل شناخت ویژگی‌های بصری برسند که به آن نقشه مشخصه[3] گفته می‌شود.[۱۹]

آن‌ها در سیستم‌های توصیه‌گر, بخش‌بندی و طبقه‌بندی تصاویر، پردازش تصاویر پزشکی، پردازش زبان طبیعی، واسط‌های کامپیوتر - مغز و سری‌های زمانی اقتصادی کاربرد دارند.

شبکه‌های عصبی کانولوشنال نسخه‌های منظم از پرسپترون[4]های چند لایه هستند. معمولاً هر نورون در یک لایه به تمامی نورون‌ها در لایه بعدی متصل می‌شود. اتصال کامل در این شبکه‌ها، آن‌ها را در معرض بیش برازش[5] داده‌ها قرار می‌دهد. شبکه‌های عصبی با الهام از فرآیندهای زیستی در نظر گرفته شدند، که الگوی اتصال بین نورون‌ها شبیه به سازمان نورونی بینایی حیوانات است. هر یک از نورون‌ها غشایی فقط یک منطقه محدود از میدان دیداری را که به عنوان میدان پذیرا شناخته می‌شود را، در بر می‌گیرند و به محرک‌ها واکنش می‌دهند. رشته‌های حساس نورون‌های مختلف تا حدی با هم همپوشانی دارند تا کل میدان دیداری را پوشش دهند.[۲۰] هابل و ویسل در دهه‌های ۱۹۵۰ و ۱۹۶۰ نشان دادند که قشای بصری گربه حاوی نورون‌ها هستند که به تنهایی به مناطق کوچکی از میدان دیداری پاسخ می‌دهند. در صورتی که چشم‌ها حرکت نکنند، منطقه‌ای از فضای بصری که در آن محرک‌های بصری بر روی یک نورون واحد تاثیر می‌گذارند، به عنوان میدان دریافت‌کننده آن شناخته می‌شود. [۲۱]

---

[1] Strain Identification by Alignment to Near Neighbors

[2] Convolutional Kernels

[3] Features map

[4] Perceptrons

[5] Overfitting

در گذشته مدل‌های سنتی پرسپترون چند لایه (MLP[1]) برای شناسایی تصویر مورد استفاده قرار گرفتند. با این حال، اتصال کامل بین گره‌ها باعث شد که عدم آنالیز داده ها در بعد های زیاد که در بعد های کمتر رخ نمی‌دهند، تکرار شود و از نظر محاسباتی برای تصاویر با وضوح بالاتر، غیرقابل انجام بود. تصویری با ۱۰۰۰ * ۱۰۰۰ - پیکسل با کانال‌های رنگی قزمز،سبز و آبی ۳ میلیون وزن دارد که پردازش در این مقیاس با اتصال کامل وزن‌ها بسیار پیچیده و ناکارامد بود. شبکه عصبی پیچشی چالش‌های ناشی از معماری پرسپترون چند لایه را با بهره‌برداری از همبستگی قوی فضایی موجود در تصاویر طبیعی, کاهش می‌دهد. باتوجه به سه تفاوت که در ادامه ذکر و مختصرا توضیح داده می-شوند:[۱۹]

۱. اتصال درون شبکه‌ای: با پیروی از مفهوم میدان‌های پذیرا، شبکه‌های عصبی پیچشی با اجرای الگوی اتصال درون شبکه‌ای بین نورون‌ها و لایه‌های مجاور، از محل فضایی گسترده‌تری بهره‌مند هستند، بنابراین این معماری قوی‌ترین فیلترهای آموزش‌دیده که واکنش بهتری به الگوی ورودی فضایی را دارند، تولید می‌کنند. پشته سازی بسیاری از این لایه‌ها منجر به فیلترهای غیر خطی می‌شود که به طور فزاینده‌ای گسترده است، به طوری که ابتدا شبکه، قطعات کوچکی از ورودی را بررسی می‌کند، سپس از آن‌ها نمایش‌های مناطق بزرگ‌تر را فراهم می‌کند.

۲. وزن‌های تسهیم شده[2]: در شبکه عصبی پیچشی، هر فیلتر در کل میدان دیداری جا به جا می‌شود. این واحدها ویژگی‌های یک‌سانی را (بردار وزن و بایاس)به اشتراک می‌گذارند و یک نقشه ویژگی را شکل می‌دهند. این بدان معنی است که تمامی نورون‌ها در یک لایه کانولوشن داده‌شده به یک ویژگی در حوزه واکنشی خاص خود، واکنش می‌دهند.[۲۲]

۳. پولینگ: در لایه‌های پولینگ، نقشه‌های ویژگی شبکه عصبی پیچشی به زیر مناطق مستطیلی تقسیم می‌شوند، مشخصه‌ها در هر مستطیل به طور مستقل از یک مقدار که معمولاً با استفاده از میانگین یا حداکثر مقدار آن‌ها به دست می‌آیند نمونه‌برداری می‌شود. علاوه بر کاهش اندازه

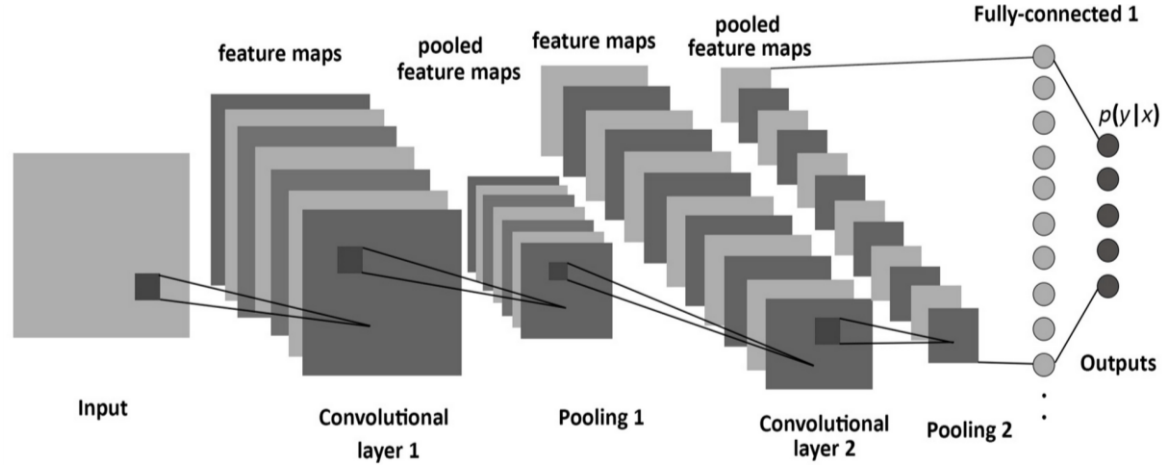
[1] Multiple Layer Perceptron

[2] Shared weights

نقشه‌های مشخصه، عملیات پولینگ از ویژگی‌های موجود در آن مستطیل کمک می‌گیرد و به شبکه اجازه می‌دهد که نسبت به تغییرات در موقعیت خود قدرتمندتر باشد.

۲-۳-۱- معماری شبکه‌های عصبی پیچشی[۲۳]

یک شبکه عصبی پیچشی از یک لایه ورودی, لایه‌های پنهان و یک لایه خروجی متشکل است. در هر شبکه عصبی جلو رونده[1], هر لایه میانی به این دلیل که ورودی‌ها و خروجی‌های آن‌ها توسط تابع فعال‌سازی و کانولوشن نهایی پوشش داده می‌شوند، پنهان می‌شوند. در یک شبکه عصبی پیچشی, لایه‌های پنهان شامل لایه‌هایی هستند که اعمال کانولوشنی را تشکیل می‌دهند. به طور معمول این پروسه شامل یک لایه است که محاسبات کرنل های کانولوشنال را با ماتریس ورودی انجام می‌دهد. وقتی کرنل کانولوشنی در امتداد ماتریس لایه ورودی حرکت می‌کند, عملیات کانولوشن، یک نقشه مشخصه(ویژگی) را ایجاد می‌کند که به نوبه خود به ورودی لایه بعدی کمک می‌کند. این پروسه توسط لایه‌های دیگر مانند لایه‌های پولینگ، لایه‌های کاملاً متصل و لایه‌های نرمال سازی دنبال می‌شود.



شکل۲- ۱ : نمایی از پیکرسازی ساختمان شبکه‌ی عصبی پیچشی[۲۳]

[1] Feed forward

در شکل۲-۱ همان‌طور که قابل مشاهده‌است ورودی به اولین لایه‌ی کانولوشنی داده شده‌است سپس بعد از استخراج ویژگی‌ها و پولینگ دوباره این عملیات تکرار شده‌است و در انتها به خروجی می‌رسد.

۲-۳-۱-۱-لایه کانولوشن

در شبکه‌ی پیچشی، ورودی یک تنسور[1] با شکلی به صورت زیر : (تعداد ورودی‌ها) * (ارتفاع ورودی) * (پهنای ورودی) *(کانال‌های ورودی) است. پس از عبور از یک لایه‌ی کانولوشنی تصویر به یک نقشه مشخصه خلاصه می‌شود که به شکل :(تعداد ورودی‌ها)*(ارتفاع نقشه ویژگی) * (پهنای نقشه) * (کانال نقشه ویژگی) است. یک لایه کانولوشنی شبکه به طور کلی ویژگی‌های زیر را دارد:

۱. فیتلر ها و کرنل‌های کانولوشن که با عرض و ارتفاع تعریف شده‌اند.

۲. تعداد کانال‌های ورودی و کانال‌های خروجی باید به درستی محاسبه شود، به صورتی که در یک لایه ورودی باید تعداد کانال‌های ورودی برابر تعداد کانال‌های خروجی (همچنین عمق نامیده می‌شود) آن باشد.

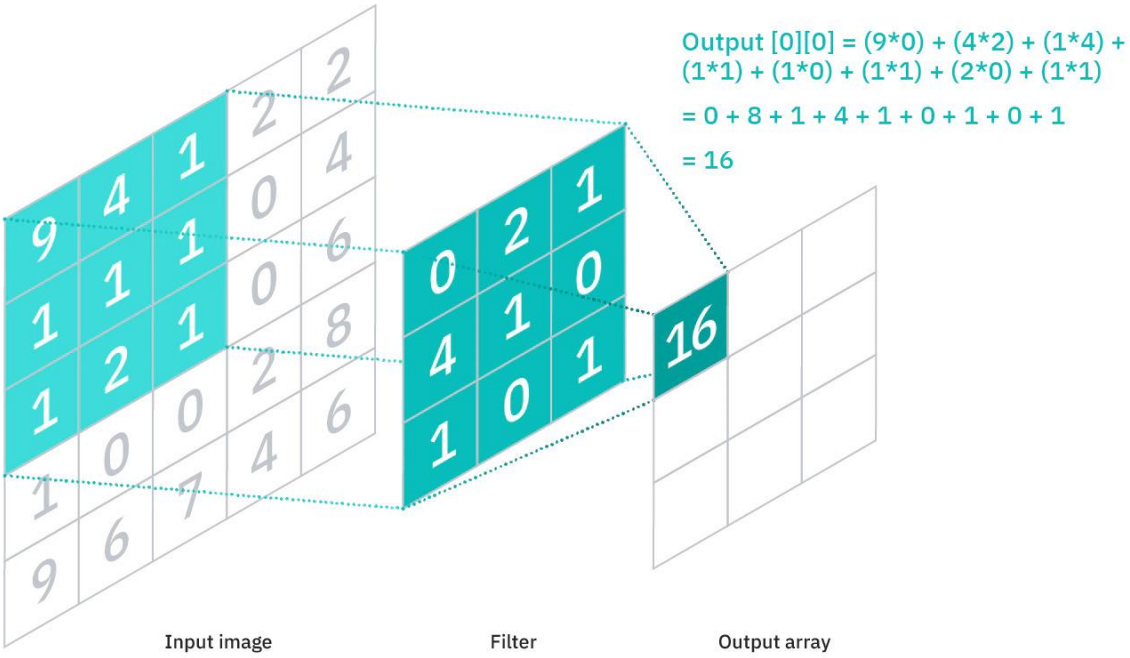
۳. دارای پارامتر های دیگر عملیات از جمله : stride, padding و dilation است.

لایه‌های کانولوشن، ورودی را در محاسبات (کانوالو[2]) دخیل می‌کنند و نتیجه آن را به لایه بعدی منتقل می‌کنند. این شبیه واکنش یک نورون در پوسته بینایی به یک محرک خاص است. هر نورون کانولوشنی داده‌ها را تنها برای حوزه پذیرای خود پردازش می‌کند. اگر چه شبکه‌های عصبی کاملاً متصل شده می‌توانند برای یادگیری ویژگی‌ها و طبقه‌بندی داده‌ها مورد استفاده قرار گیرند, این معماری به طور کلی برای ورودی‌های بزرگ‌تر مانند تصاویر با وضوح بالا، غیرعملی است. این امر نیازمند تعداد بسیار بالایی از نورون‌ها است. به عنوان مثال, یک لایه کاملاً متصل برای یک تصویر (کوچک) با اندازه‌ی ۱۰۰ *۱۰۰ برای هر نورون در لایه دوم دارای ۱۰۰۰۰ وزن است. در عوض, کانولوشن تعداد پارامترهای آزاد را

[1] Tensor

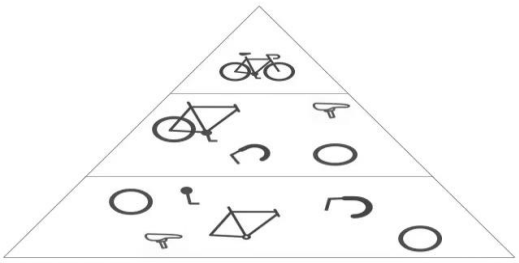
[2] convolve

کاهش می‌دهد و به شبکه امکان می‌دهد که عمیق‌تر شود. با توجه به دلیل مطرح شده برای یادگیری تصاویر، اکثرا از این شبکه استفاده می شود که هم دقت و هم کارامدی در آن بیشتر است.[۲۳،۲۴]



شکل۲- ۲ یک مثال از عمکرد لایه کانولوشن[۲۴]

همانطور که در تصویر بالا شکل ۲ می‌بینید، هر مقدار خروجی در نقشه ویژگی لازم نیست به هر مقدار پیکسل در تصویر ورودی متصل شود. تنها لازم است که به فضای پذیرا، جایی که فیلتر در آن اعمال می‌شود، متصل شود. از آنجا که آرایه خروجی نیازی به نگاشت مستقیم به هر مقدار ورودی ندارد، لایه‌های کانولوشن و پولینگ عموماً به لایه‌های نسبی منسوب هستند. با این حال، این ویژگی نیز می‌تواند به عنوان ویژگی های عمیق توصیف شوند.



شکل۲- ۳ : چگونگی رفتار لایه های متوالی کانولوشن[۲۴]

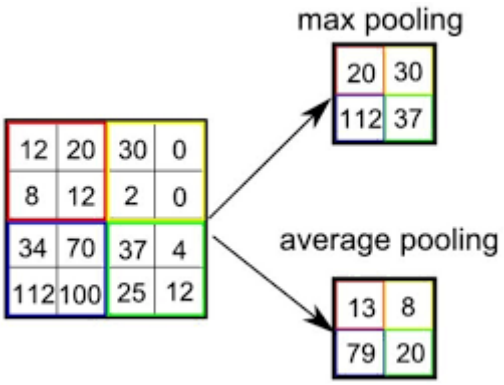
در شکل ۲-۳ چگونگی جداسازی ویژگی‌های یک تصویر را به صورت شماتیک برای شناسایی یک المان نشان می‌دهد. یک لایه کانولوشن دیگر می‌تواند لایه کانولوشن اولیه دنبال کند. هنگامی که این اتفاق می‌افتد، ساختار شبکه می‌تواند به صورت سلسله مراتب رفتار کند، چون لایه‌های بعدی می‌توانند پیکسل‌های درون لایه‌های اولیه را ببینند. به عنوان مثال، اجازه دهید فرض کنیم که ما در تلاشیم تا مشخص کنیم که آیا یک تصویر دارای یک دوچرخه است یا نه. شما می‌توانید به دوچرخه به عنوان یک مجموع قطعات فکر کنید. شامل یک ساختمان اصلی، فرمان، چرخ‌ها، پدال‌ها و غیره است. هر بخش از دوچرخه یک الگوی سطح پایین‌تر را در شبکه عصبی ایجاد می‌کند و ترکیب بخش‌های آن یک الگوی سطح بالاتر و عمیق‌تر را نشان می‌دهد که یک سلسله‌مراتب ویژگی را در شبکه ایجاد می‌کند.

۲-۳-۱-۲-لایه پولینگ [۲۵]

یکی دیگر از مفاهیم مهم شبکه عصبی پیچشی تجمع یا پولینگ است که نوعی از نمونه‌گیری غیر خطی است. چندین تابع غیر خطی برای اجرای پولینگ وجود دارند که در آن پولینگ حداکثرگیری[1] بیش‌ترین استفاده را دارد. در این عملیات تصویر ورودی را به مجموعه‌ای از مستطیل‌ها تقسیم می‌کند و در هر ناحیه فرعی تولید شده، حداکثر را خروجی می‌دهد. این ایده مبنای استفاده از پولینگ در شبکه‌های عصبی کانولوشنال است. لایه پولینگ به منظور کاهش تدریجی اندازه فضایی نمایش در راستای کاهش تعداد پارامترها, حافظه‌ی استفاده شده و مقدار محاسبات در شبکه است و از

[1] Max pooling

این رو به بیش‌برازش را نیز کنترل می‌کند . این روش به عنوان نمونه‌برداری استفاده می‌شود و بـه طـور مرحله‌ای یک لایه بعد از لایه‌های متوالی کانولوشنال در معماری شبکه عصبی ایجاد می‌گردد(هر یـک از آن‌ها معمولاً با یک تابع فعال‌ساز[1] مانند ReLU همراه هستند.). به غیر از پولینـگ حـداکثرگیر، نـوعی دیگر از پولینگ که میانگین‌گیر[2] نامیده شده‌است نیز رایج است. هنگامی که فیلتر میـان ورودی حرکـت می‌کند، مقدار متوسط درون یک فیلد دریافت‌کننده را محاسبه می‌کند تا به آرایه خروجی ارسال کند.



شکل۲- ۴ مثالی از عملکرد دو نوع پولینگ[۲۵]

در شکل۲-۴ دو نوع پولینگ قابل مشاهده‌است، در بالا حداکثرگیر و در پایین میانگین‌گیر است. در حالی که بسیاری از اطلاعات در لایه پولینگ از دست رفته‌است، اما این عملیات دارای تعدادی زیـادی مزایای برای شبکه عصبی کانولوشنال نیز هست. پولینگ‌ها به کاهش پیچیدگی، بهبود کارایی و محـدود کردن ریسک اورفیتینگ[3] کمک می‌کنند.

[1] Activation function

[2] Average pooling

[3] Overfitting

۲-۳-۱-۳-لایه های کاملا متصل[1]

همانطور که از نام این لایه ها مشخص است، دراین لایه‌های کاملاً متصل، هر گره در لایه خروجی مستقیماً به یک‌گره در لایه قبلی متصل می‌شود. این لایه وظیفه طبقه‌بندی براساس ویژگی‌های استخراج‌شده از لایه‌های قبلی و فیلترهای مختلف آن‌ها را انجام می‌دهد. در حالی که لایه‌های کانولوشنال و پولینگ گرایش به استفاده از توابع رلو[2] دارند، لایه‌های کاملا متصل معمولاً از یک تابع فعال سافت‌مکس برای طبقه‌بندی ورودی‌های مناسب استفاده می‌کنند و یک احتمال از ۰ تا ۱ تولید می‌کنند.[۲۵]

۲-۳-۲- توابع فعال‌ساز[3]

تابع فعال‌ساز یک‌گره است که در انتهای یا بین شبکه‌های عصبی قرار داده می‌شود. تابع فعال-ساز تبدیلی است که ما در سیگنال ورودی اعمال می‌کنیم. این خروجی تغییر یافته سپس به لایه بعدی نورون‌ها به عنوان ورودی فرستاده می‌شود. توابع فعال‌ساز می‌توانند خطی یا غیرخطی باشند. توجه شود که انتخاب یک تابع فعالساز مناسب، عملکرد محاسباتی شبکه‌های عصبی را بهبود می‌بخشد. از آنجایی که شبکه‌های عصبی، نتایج خطی را از نگاشت‌ها تولید می‌کنند، نیاز است تا توابع فعال‌ساز، به منظور محاسبات پیشرو و به خصوص یادگیری الگوهای موجود در داده‌ها، این خروجی‌های خطی را به خروجی‌های غیرخطی تبدیل کنند. توجه شود که خروجی‌های هر لایه، در شبکه‌های چند لایه مانند شبکه‌های عصبی عمیق، به لایه‌های بعدی رفته تا خروجی نهایی به دست بیایند. خروجی مورد نظر نیز نوع تابع فعال‌سازی که احتیاج است را مشخص می‌کند. این توابع برای تبدیل سیگنال‌ها و مدلهای خطی ورودی به سیگنال‌های غیرخطی خروجی مورد استفاده قرار می‌گیرند. همچنین به یادگیری بهتر چندجمله‌ای‌های مرتبه بالا در شبکه های عمیق نیز کمک می‌کنند.[۲۶]

---

[1] Fully-connected layers

[2] ReLU

[3] Activation function

خاصیت ویژه‌ی توابع فعال‌ساز غیرخطی، مشتق پذیر بودن آن‌ها است که در صورت نبود این ویژگی، این توابع قادر به کار کردن در حین عملیات پس‌انتشار در شبکه‌های عصبی عمیق نیستند. در ادامه به معرفی بعضی از توابع معروف و پرکاربرد فعال‌ساز می پردازیم.

۲-۳-۲-۱-تابع فعال‌ساز سیگموید[1][۲۶]

تابع سیگموید یک تابع ریاضی با منحنی شبیه به حرف "S" است یه به آن منحنی سیگموئیدی نیز گفته شده است و e در فرمول عدد اویلر است. این تابع به معادلات زیر نمایش داده می‌شود:

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - S(-x).$$

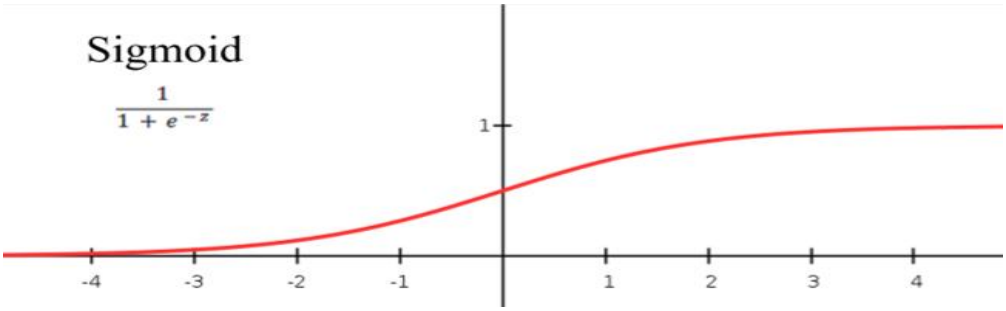
مجموعه وسیعی از توابع سیگموید شامل توابع لوجستیک[2] و هایپربولیک[3] به عنوان توابع فعال‌ساز نورون‌های در شبکه عصبی استفاده شده‌است. منحنی‌های سیگموید نیز در آمار به عنوان توابع توزیع تجمعی رایج هستند (که از ۰ تا۱ مقداردهی می‌شوند). این توابع در کاربردهایی مانند انتگرال تراکم لوجستیک، چگالی نرمال و چگالی احتمال متداول هستند. تابع سیگموید منطقی مشتق‌پذیر است و معکوس آن تابع لاجیت[4] است.

[1] Sigmoid

[2] logistic

[3] hyperbolic

[4] logit

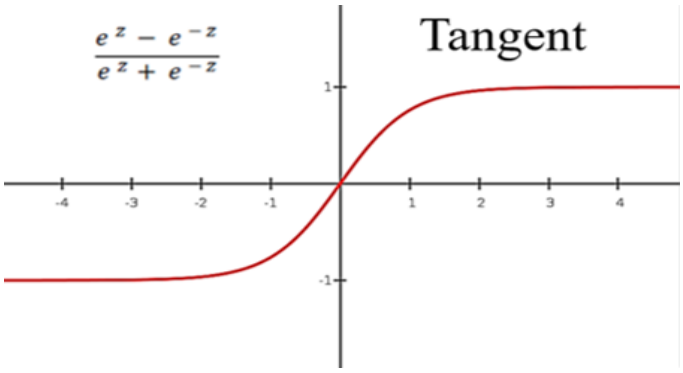


شکل۲-۵ تابع فعال‌ساز سیگموید[۲۶]

۲-۳-۲-۲-تابع فعال‌ساز تانژانت[1]

تابع فعال‌ساز هایپربولیک تانژانت به سادگی به عنوان تابع Tanh (همچنین "tanh" و "TanH") نیز منسوب است. عملکرد این تابع بسیار شبیه عملکرد تابع فعال‌ساز سیگموید است و حتی همان شکل S مانند را دارد. این تابع هر مقدار حقیقی را به عنوان ورودی و خروجی در محدوده[−۱،۱] می‌برد. هرچه ورودی بزرگ‌تر داده گردد، مقدار خروجی به ۱ نزدیکتر خواهد بود، در حالی که هرچه ورودی کوچک‌تر (منفی تر) بود، خروجی نزدیک‌تر به ۱- خواهد بود.e در فرمول عدد اویلر است.[۲۷]

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$$



شکل۲- ۶ تابع فعال‌ساز تانژانت[۲۷]

در شکل ۲-۶ قادر به مشاهده نمودار تانژانت است.

[1] tangent

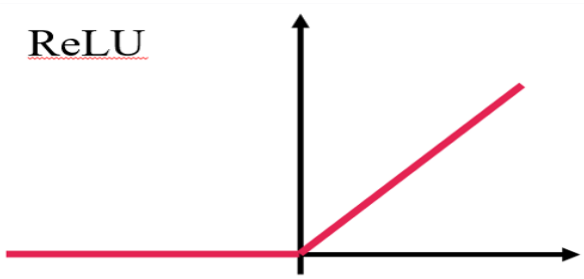
۲-۳-۲-۳-تابع فعال‌ساز رلو (ReLU[1])

در زمینه شبکه‌های عصبی، تابع فعال کننده رلو یا ReLU یک تابع فعال‌ساز است که به عنوان بخش مثبت آرگومان آن تعریف شده‌است و معادله آن به صورت زیر نمایش داده شده‌است:

$$f(x) = x^+ = \max(0, x)$$

که در آن x ورودی نورون است. این تابع هم چنین به عنوان یک تابع شیب شناخته می‌شود و شبیه به rectification half-wave در مهندسی برق است. این تابع فعال‌ساز در زمینه استخراج ویژگی تصویری در سلسله مراتبی از شبکه‌های عصبی که از اواخر دهه ۱۹۶۰ شروع شد، اولین بار دیده شد. بعدها مشخص شد پتانسیل‌های بیولوژیکی قوی و توجیه ریاضی دارد. در سال ۲۰۱۱ مشاهده شد که آموزش بهتر شبکه‌های عمیق‌تر، در مقایسه با توابع فعال فعال‌ساز قبل از سال ۲۰۱۱، به عنوان مثال، سیگموید (که از تیوری احتمال الهام‌گرفته است)و همتای آن کع تابع تانژانت است، امکان‌پذیر است. تایع فعال‌ساز رلو ، تا سال ۲۰۱۷، محبوب‌ترین تابع فعالساز برای شبکه‌های عصبی عمیق بودند. [۲۷،۲۸]

[1] Rectified Linear Unit



شکل۲- ۷ تابع فعال‌ساز رلو[۲۷]

در شکل ۲-۷ نمودار تابع فعال‌ساز رلو تریسم شده‌است.

۲-۴- کاربردهای شبکه‌های عصبی کانولوشنی

۲-۴-۱- تشخیص تصاویر

این نوع شبکه‌ها اغلب در سیستم‌های تشخیص تصاویر مورد استفاده قرار می‌گیرند. در سال ۲۰۱۲، نرخ خطای ۰/۲۳ درصد در پایگاه‌داده MNIST گزارش شد. در مقاله‌ی Flexible, High "Performance" Convolutional Neural Networks for Image Classification [۲۹] در مورد استفاده شبکه‌های عصبی کانولوشنی برای طبقه‌بندی تصویر گزارش داد که فرآیند یادگیری به طرز شگفت آوری سریع است؛ در این مقاله، بهترین نتایج منتشر شده در سال ۲۰۱۱ در پایگاه‌داده MNIST و پایگاه‌داده NORB بدست‌آمده است. پس از آن، یک شبکه مشابه به نام الکس‌نت به عنوان پیشتاز در چالش شناخت بصری سال ۲۰۱۲ برنده شد. این شبکه‌ها هنگامی که برای تشخیص چهره به کار برده شدند، به کاهش زیادی در نرخ خطا دست یافتند .در شناسایی ۵۶۰۰ تصویر در ۱۰ دسته مختلف به دقتی برابر۹۷/۶ درصد دست یافتند. از این شبکه عصبی نیز برای ارزیابی کیفیت ویدئو در یک روش بصری بعد از آموزش دستی استفاده شد و به خطای جذر میانگین مربعات کوچکی منجرشد.[۲۲] در سال ۲۰۱۵، چند شبکه عصبی کانولوشنی با لایه‌های زیاد با عملکرد رقابتی توانایی تشخیص چهره‌ها از طیف وسیعی از زوایا را نشان دادند، از جمله وارونه و حتی زمانی که تا حدودی مسدود شده‌اند. این شبکه‌ها بر روی یک پایگاه داده از ۲۰۰،۰۰۰ تصویر آموزش داده شدند که شامل صورت‌های مختلف و

جهت گیری های مختلف و ۲۰ میلیون تصویر دیگر که چهره‌ای در آن‌ها نبود. آن‌ها از بچ[1]هایی۱۲۸تایی از تصاویربا بیش از ۵۰۰۰۰ تکرار استفاده کردند.[۳۰]

۲-۴-۲- تشخیص ویدیو

در مقایسه با دامنه‌های پژوهش روی داده‌های تصویری، کار نسبتا کمی برای کاربرد این شبکه در طبقه‌بندی ویدئو وجود دارد. ویدیو پیچیده‌تر از تصاویر است چون بعد (زمان)دیگری دارد. با این حال، محاسبات کانولوشنی جدیدی در حوزه ویدیو برای حل تفاوت بعدی فضا و زمان به عنوان ابعاد معادل ورودی کشف شده‌اند و اجرای شبکه های عصبی کانولوشنی برای زمان و فضا به طور هم‌راستا امکان‌پذیر شد. راه دیگر ترکیب ویژگی‌های دو شبکه عصبی کانولوشنی، یکی برای فضایی و مکانی ودیگری برای زمان است. طرح‌های یادگیری بدون نظارت برای آموزش سری زمانی معرفی شد، که براساس کانولوشن بولتزمن و تحلیل ساب‌سپیس[2] به طور مستقل تعریف شده‌است.

۲-۴-۳- پردازش زبان طبیعی

این شبکه‌ها نیز برای پردازش زبان طبیعی مورد بررسی قرار گرفته‌اند. مدل‌های از آن برای مشکلات مختلف NLP[3] موثر هستند و به نتایج عالی در تجزیه معنایی، مدل‌سازی جمله، طبقه‌بندی، پیش‌بینی و سایر عملکردهای معمول NLP دست یافته‌اند.

[1] Batch

[2] Subspace

[3] Neuro-linguistic programming

فصل سوم

شناسایی احساسات و چهره

۳-۱- شبکه عصبی و احساسات

در دنیای مدرن، سیستم‌های هوش مصنوعی به دلیل توانایی‌های غنی و استفاده‌های کارآمد خود، محبوبیت زیادی به دست آورده‌اند. در حال حاضر، شبکه‌های عصبی به طور فعال در سیستم‌های هوش مصنوعی مورد استفاده قرار می‌گیرند. روش‌های بیومتریک مبتنی بر هوش مصنوعی به شما این امکان را می‌دهد که نه تنها فیزیک چهره انسان‌ها را بلکه ویژگی‌های رفتاری افراد را شناسایی، آنالیز و اندازه‌گیری کنید. آن‌ها به راحتی می‌توانند به شناسایی احتمال هر گونه قانون‌شکنی از جمله انواع دزدی‌ها مانند سرقت از بانک، به نیروهای امنیتی محلی قبل از این که این حادثه رخ دهد کمک کنند. این فناوری می‌تواند به صورت موازی با تحلیل متن[1] و پردازش زبان[2] طبیعی کار کنند. تجزیه و تحلیل‌های عاطفی یک ترکیب جالب از روان‌شناسی و فن‌آوری است. یکی از راه‌های اصلی برای تشخیص احساسات، آنالیز حالات چهره است. عواطف پایه و اصلی از لحاظ بیولوژیکی ثابت و ذاتی در نظر گرفته می‌شوند و در نتیجه برای همه مردم و همچنین برای بسیاری از حیوانات، همگانی محسوب شده است.

عواطف پیچیده، ترکیبی از مجموعه‌ای از احساسات پایه و اصلی هستند و یااحساسات ویژه و غیرعادی خوانده می‌شود. مشکل اصلی، تعیین این است که کدام عواطف پایه هستند وکدام احساسات پیچیده هستند. براساس مطالعات روانشناسی اطلاعات تصویری ادراک و جهت‌دهی گفتار را تغییر می‌دهد، می‌توان فرض بر این کرد که ادراک عواطف انسانی از یک روند مشابه پیروی می‌کند.

توابع تشخیص چهره، حالت‌های چهره را در ویدئوها و عکس‌ها بررسی می‌کند و اطلاعاتی نسبی را نشان می‌دهد که احساسات عمومی مثل شادی، ناراحتی، عصبانیت، ترس، تعجب، تنفر و حالت طبیعی را تعریف می‌کنند.

[1] Text analysis

[2] Natural language processing

۳-۱-۱- انواع احساسات [۱،۲]

احساسات برای انسان‌ها و حیوانات در تعریف‌های فلسفه ای و روانشناختی به دو دسته‌ی کلی تقسیم می‌شوند:

۱. احساسات پایه

۲. احساسات پیچیده

احساسات پایه به هفت احساس گفته می شوند که برای عموم افراد تعریف شده و درک‌پذیر است و با ویژگی های مبرا از هم، قابل تعریف هستند. احساسات پیچیده به طور کلی به دسته‌ای از احساسات تلقی می‌گردد که یا ترکیبی از احساسات پایه هستند یا در دسته‌ی خاصی از افراد دیده شدند و به احساسات ویژه و ناهنجار نام گذاری شده‌اند. در ادامه به تعریف هفت احساس اصلی می پردازیم.[۱،۲]

۱. خوشحالی: شایع‌ترین بیان احساسی که یک فرد در زندگی عادی برای بروز خرسندی از پیامدی نشان می‌دهد. در فیزیولوژی، لبخند، حالتی از چهره است که وقتی ماهیچه‌ها به خصوص در هر دو طرف دهان به سمت گونه‌ها خم می‌شوند.

۲. ناراحتی: احساسی تلخ مربوط به پیامدهای آزاردهنده مانند از دست دادن خواسته، درد و درماندگی است.

۳. عصبانیت: عصبانیت یا خشم یک واکنش طبیعی به تهدیدات خاص است. در نتیجه، پرخاشگری نوعی از واکنش مرتبط به خشم است.

۴. ترس: اطلاعات مربوط به خطر قریب‌الوقوع، تهدید قریب‌الوقوع و تمایل به فرار را بیان می‌کنند.

۵. تعجب: واکنشی است که تقریباً همیشه در پاسخ به رویدادی پیش‌بینی‌نشده رخ می‌دهد و پیامی حاوی یک چیز ناگهانی، جدید و غیر منتظره را منتقل می‌کنند.

۶. تنفر: حالت انزجار یا تنفر اغلب به عنوان تفسیری از زجر بردن در بسیاری از رویدادها و افرادی که سبب واکنش‌های منفی می‌شوند، نمایش داده می‌شود.

۷. طبیعی: به حالتی خنثی و عاری از احساسات پایه گفته می شود.

۳-۱-۲ کاربردهای فناوری تشخیص احساسات[۲]

دوربین‌های امنیتی برای شناسایی مجرمان در حال ارتکاب جرم مورد استفاده قرار می‌گیرند. با این حال، در حال حاضر، شرکت‌ها به دنبال استفاده از دوربین‌های نظارتی برای نظارت بر افراد برای تحقیق در زمینه‌ی تبلیغات هستند. در آلمان، توسعه دهندگان دوربین‌هایی را در تبلیغات خیابانی جای-گذاری می‌کنند و تلاش برای تعیین عکس‌العمل ایموشنی مردم نسبت به تبلیغات دارند. استفاده پژوهشگران هلندی از نرم‌افزارهای شناسایی احساسات در کامپیوترها و تلفن‌های همراه برای آزمایش واکنش‌های مردم به تبلیغات و بازاریابی رایج است.

مثال‌هایی از استفاده از تکنولوژی تشخیص احساسات عبارتند از:

- شناسایی یک مهاجم بالقوه براساس آنالیز احساسات او
- تجزیه و تحلیل اثربخشی تبلیغات
- تعیین اثربخشی روش برای فروش کالا در یک فروشگاه
- تست در مقیاس بزرگ در حوزه واکنش به محصولات (واکنش چهره)

۳-۲- دیتاست[1]های تشخیص احساسات

یک دیتاست احساسات صورت، مجموعه‌ای از تصاویر و یا کلیپ های ویدئویی با حالت‌های چهره از احساسات مختلف است. محتوای این مجموعه برای آموزش، آزمایش و اعتبار سنجی الگوریتم‌ها برای توسعه سیستم‌های تشخیص احساسات ضروری است. تفسیر احساسات دیتاست می‌تواند در برچسب‌های احساسی گسسته و یا در مقیاس پیوسته انجام شود. اکثر دیتاست‌ها معمولا بر پایه‌ی تئوری احساسات پایه و اصلی (نوشته اکمن) هستند که وجود شش احساس اصلی مجزا را فرض می‌کند (عصبانیت، ترس،

[1] Dataset

تنفر، شادی، تعجب و ناراحتی). با این حال، برخی از دیتاست‌هاشامل برچسب گذاری احساسی در مقیاس بزرگ‌تر و وسیع تر هستند.

در دیتاست‌هایی که ژست گرفته‌شده نامیده شدند، از شرکت‌کنندگان خواسته می‌شود حالات احساسی ابتدایی مختلفی را نمایش دهند، در حالی که در دیتاست حالات خود به خودی شرکت-کنندگان، در حالات طبیعی هستند. حالات خود به خودی از لحاظ شدت، پیکربندی و مدت متفاوت هستند. بنابراین، در اغلب موارد، تصاویر ژست دهی‌شده اغراق‌آمیز هستند، در حالی که حرکات خود به خودی، ظریف هستند و در ظاهر متفاوت هستند.

۳-۲-۱- معرفی تعدادی از دیتاست های مرتبط[۳۲]

گستره زیادی از دیتاست‌ها برای تشخیص احساسات در دهه‌ی اخیر منتشر شده اند. البته تفاوت‌های زیادی در رابطه با حیطه‌ی کاری آن‌ها وجود دارد. اولین و تاثیرگذارترین مشخصه نحوه‌ی طبقه‌بندی دیتاست‌ها یا همان برچسب‌گذاری آن‌ها است. اکثرا از تئوری اکمن در طبقه‌بندی احساسات استفاده شده‌است اما در بعضی از دیتاست‌ها بعضی از آن‌ها حذف شده‌اند و یا حتی بعضی از احساسات فرعی به آن‌ها اضافه شده‌است. برای مثال دیتاست FEI برای فقط تشخیص لبخند در چهره فراهم شده-است. تعداد تصاویر یا ویدیوهای داخل دیتاست که جهت آموزش شبکه مورد استفاده قرار می‌گیرند، شاخصه دیگر در انتخاب مناسب‌ترین دیتاست برای هر پروژه است. البته باید در این موضوع به برچسب گذاری داده‌ها و صحت آن توجه لازم را داشت، زیرا ممکن است در دیتاستی با داده های زیاد، امکان نبود برچسب برای تعدادی از داده‌ها یا برچسب‌گذاری غلط روی آن‌ها هست که منجر به آموزش بدون دقت شبکه شود و نهایتا مدل قادر به جواب‌دهی صحیح نیست. در دیتاست نیز کیفیت[1] تصاویر از ویژگی‌های حائذ اهمیت است، زیرا داده‌های کم کیفیت نه تنها استفاده مثبتی در مدل‌سازی ندارند حتی ممکن است، از دقت سیستم بکاهند. در ادامه جدولی از دیتاست‌های تشخیص احساسات را مشاهده می‌کنید.

عنوان دیتاست	طبقه‌بندی احساسات	تعداد تصویر/ویدیو	کیفیت داده	طوسی/رنگی

[1] Resolution

اکثرا طوسی	۴۹۰ * ۶۴۰	۵۹۳ عکس (۳۲۷ داده با برچسب مشخص)	خوشحالی، ناراحتی، خشم، ترس، تنفر، تعجب،تحقیر، طبیعی	Extended Cohn-Kanade Dataset (CK+)
طوسی	۲۵۶*۲۵۶	۲۱۳ عکس	خوشحالی، ناراحتی، خشم، ترس، تنفر، تعجب، طبیعی	Japanese Female Facial Expressions (JAFFE)
رنگی	۸۹۶*۸۹۶	۱۴۶۲ عکس و ویدیو	خوشحالی، ناراحتی، خشم، ترس، تنفر، تعجب، طبیعی	Multimedia Understanding Group (MUG)[
رنگی	۱۰۲۴*۶۸۱	۸۰۴۰ عکس	خوشحالی، ناراحتی، خشم، ترس، تنفر، تعجب،تحقیر، طبیعی	Radboud Faces Database (RaFD)
رنگی	۴۸۰*۶۴۰	۲۸۰۰ عکس	خوشحالی ، طبیعی	FEI Face Database
طوسی	۴۸۰*۴۸۰	۳۵۸۸۷ عکس	خوشحالی، ناراحتی، خشم، ترس، تنفر، تعجب، طبیعی	FER-2013
متفاوت	متفاوت	۵۰۰۰۰۰ عکس (برچسب-گذاری بدون صحت)	خوشحالی، ناراحتی، خشم، ترس، تنفر، تعجب،تحقیر، طبیعی	AffectNet
رنگی	متفاوت	۲۹۶۷۲ عکس و ویدیو	خوشحالی، ناراحتی، خشم، ترس، تنفر، تعجب،تحقیر، طبیعی(همراه ۲۶ احساس فرعی)	iReal-world Affective Faces Database (RAF-DB)

جدول ۱-۱ تعدادی از دیتاست‌های تشخیص احساسات

۳-۳- دیتاست FER[1]

دیتاست FER که توسط ککل[2] جمع آوری شده‌بود در سال ۲۰۱۳ در کنفرانس بین‌المللی توسط پیر لوک کریر و آرون کورویل یادگیری ماشین رونمایی شد. در دیتاست صورت ها به هفت دسته مطرح شده در قبل دسته‌بندی شده‌اند. این تصاویر از عکس‌هایی ۴۸ پیکسل در ۴۸ پیکسل تشکیل شده‌اند. این تصاویر سیاه و سفید، چهره افراد را پوشش می‌دهند. چهره را قسمتی از صورت در آناتومی بدن تعریف کرده‌اند که از قسمت فوقانی به بالارفتن ابروها در پیشانی محدود می‌شود و در انتها به قوس لب پایین منتهی می‌شود. در راستای استفاده از دیتاست FER باید تشخیص چهره نیز لحاظ شود. این دیتاست از ۳۵۸۸۷ داده که در هفت طبقه دسته‌بندی شده‌اند، تشکیل شده‌است. این دیتاست دارای دو ردیف هست که یک ردیف دارای اعداد ۰ تا ۶ می باشد به این ترتیب که ۰-عصبانی، ۱-تنفر، ۲-ترس، ۳- خوشحال، ۴- ناراحت، ۵- متعجب و ۶-طبیعی است، که ستون احساسات است (emotion column).ستون دیگر تصویر استخراجی را به صورت برداری از اعداد بین ۰ تا ۲۵۵ مشخص می کند و به بصورت آرایه در خود نگه می‌دارد. به این ستون پیکسل نام داده اند. (pixel column) در ادامه نحوی تقسیم بندی داده ها تشریح شده‌است. [۳۲]

جدول ۱-۲ تقسیم بندی داده‌ها در FER

عنوان برچسب	تعداد داده‌های برچسب	درصد تشکیل دهنده دیتاست
عصبانی	۴۹۵۳ تصویر	٪ ۱۳/۸
تنفر	۵۴۷ تصویر	٪ ۱/۸
ترس	۴۰۹۷ تصویر	٪ ۱۱/۴۱
خوشحال	۸۹۸۹ تصویر	٪۲۵/۰۶
ناراحت	۶۰۷۷ تصویر	٪ ۱۶/۹۳
متعجب	۴۰۰۲ تصویر	٪ ۱۱/۱۰

[1] Facial Expression Recognition

[2] Kaggle

٪ ۱۷/۲۷	۶۱۹۸ تصویر	طبیعی

شکل۳-۱ تعدادی مثال از عکس های دیتاست FER-2013

در شکل۳-۱ نعدادی از تصاویر در دیتاست FER قابل مشاهده است که تنوع در سن و جنسیت را نمایش می دهد. خاکستری بودن و اندازه‌ی یکسان تصاویر در این تصویر واضح است.

۳-۳-۱- بازنگری تصاویر

در این بخش تصاویر دیتاست در نواحی مطرح شده مورد بررسی قرار گرفته‌اند. این عملیات برای افزایش صحت دیتاست فراهم شده‌است. قسمتی به صورت دستی و قسمتی به صورت رایانه‌ای انجام شده‌است.

۱. قابلیت شناسایی چهره: در این مرحله با توجه به اینکه آیا مدل آماده‌ی هار که در ادامه به توضیح آن پرداخته‌ایم، می‌تواند چهره داخل تصاویر را تشخیص دهد یا خیر، جداسازی تصاویر استاندارد انجام شد.

۲. کیفیت تصاویر: تصاویر تار و ناواضح به صورت دستی با استفاده از تبدیل ماتریس به تصویر انجام شد.

۳. صحت طبقه‌بندی: این مرحله بر دقت مدل بسیار تاثیرگذار است. این بخش توسط دانشجویان زیادی در کشورهای مختلف بررسی شده‌است به همین دلیل این دیتاست دقت قابل قبولی برای مدل فراهم می‌کند.

۴. اندازه‌ی تصاویر: در این مرحله تصاویر گلچین شده به اندازه مشخصی تغییر اندازه داده می‌شوند.

۳-۳-۲- بهینه‌سازی دیتاست

در زمینه افزایش بهینگی دیتاست روش های مختلفی مطرح شده‌است. در این پروژه در این باب با توجه به اینکه صحت و دقت دیتاست با توجه به چهار مرحله‌ی ذکر شده در عنوان قبلی قابل تایید است، سعی به گسترش تنوع جای‌گیری چهره کرده‌ایم.

برای این کار در استفاده از دیتاست به چند روش مختلف سعی بر افزایش تعداد تصاویر نه در ماهیت بلکه در نحوه نمایش داشته‌ایم. از انواع این روش‌ها می‌توان به زوم کردن در عکس، قرینه یا به اصطلاح آیینه کردن تصویر و چرخاندن عکس در زوایای مختلف اشاره کرد. با این عملیات احتمال جای-گیری مختلف چهره برای یادگیری بهتر مدل عصبی فراهم می‌شود.

۳-۴- چارچوب تشخیص اشیا ویولا-جونز

چارچوب تشخیص شئ ویولا-جونز یک چارچوب شناسایی اشیا است که در سال ۲۰۰۱ توسط پاول ویولا و مایکل جونز پیشنهاد شد. اگر چه این چارچوب می‌تواند برای تشخیص انواع کلاس‌های شئ آموزش داده شود، اما در درجه اول با مشکل تشخیص چهره مواجه است.

مساله‌ای که نیاز به حل شدن دارد تشخیص چهره‌ها در یک تصویر است. یک انسان می‌تواند این کار را به راحتی انجام دهد, اما یک کامپیوتر به دستورالعمل و محدودیت‌های دقیق نیاز دارد. برای اینکه این مهارت بیشتر قابل مدیریت باشد، ویولا-جونز به چهره‌های بدون زاویه‌گیری از جلو نیاز دارد. بنابراین برای شناسایی, کل صورت باید به سمت دوربین اشاره داشته باشد و نباید کج شود. در حالی که به نظر می‌رسد این محدودیت‌ها می‌تواند کاربرد الگوریتم را تا حدی کاهش دهد, چون گام تشخیص اغلب به

دنبال گام شناسایی است. پس سعی بر پیشرفت این تئوری شد تا مشکلات عملی‌سازی بهبود بخشد.[۳۳]

۳-۴-۱- انواع ویژگی ها و ارزیابی[۳۳]

ویژگی‌های الگوریتم ویولا-جونز که آن را به یک الگوریتم تشخیص خوب تبدیل می‌کند عبارتند از:

۱. مقاومت: نرخ تشخیص بسیار بالا (نرخ واقعی مثبت) و نرخ کاذب مثبت بسیار پایین است.

۲. زمان واقعی: برای کاربردهای عملی به حداقل ۲ فریم در ثانیه برای پردازش نیازمند است.

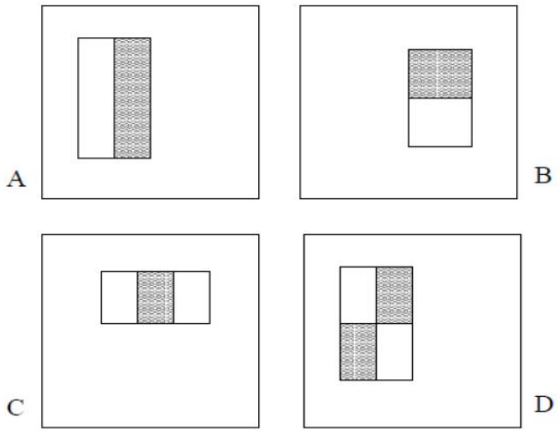
۳. تشخیص چهره تنها (بدون شناسایی[1]): هدف تشخیص چهره افراد از هر شئ غیر صورت (تشخیص اولین قدم در فرآیند شناسایی است.).

روش تشخیص هدف، تصاویر را بر مبنای ویژگی‌های ساده‌ای طبقه‌بندی می‌کند. انگیزه‌های بسیاری برای استفاده از این مشخصه‌ها به جای استفاده مستقیم از پیکسل ها وجود دارد. یک دلیل بدیهی این است که ویژگی‌ها می‌توانند برای شناسایی یک دامنه از دانش عمل کنند که برای یادگیری از متغیرهای محدود بهتر عمل می‌کنند. برای این سیستم هم چنین انگیزه‌ی دیگری وجود دارد. این انگیزه بیان‌گر این واقعیت است که سیستم های بر پایه ویژگی در زمینه‌ی سرعت از سیستم‌های بر پایه‌ی پیکسل بسیار سریع‌تر هستند. ویژگی‌هایی که به دنبال این چارچوب تشخیص به طور همگانی مورد استفاده قرار می‌گیرند, شامل مقادیر پیکسل های تصویر در نواحی مستطیلی هستند. به این ترتیب, آن‌ها شباهت زیادی به توابع پایه هار دارند. با این حال, از آنجایی که ویژگی‌های مورد استفاده ویولا-جونز به بیش از یک منطقه مستطیل‌شکل وابسته هستند, معمولاً پیچیده‌تر هستند.

[1] Recognition

۳-۴-۲- مستطیل‌های ویژگی[1]

استفاده از ویژگی‌های ساده, یادآور مبنای هار است. توابعی که توسط پاپاگورجیو و همکارانش استفاده شده‌اند. به طور خاص، ما از سه نوع ویژگی استفاده می‌کنیم. مقدار ویژگی دو مستطیل در حقیقت تفاوت بین مجموع پیکسل در دو منطقه مستطیل شکل است. مناطق مستطیلی دارای اندازه و شکل یکسان هستند و به صورت افقی یا عمودی قرار دارند. ویژگی سه مستطیل به محاسبه تفاوت مجموع سطح دو مستطیل خارجی از مجموع سطح مستطیل مرکزی می‌پردازد. در نهایت ویژگی چهار مستطیل است که این ویژگی تفاوت قطری بین زوج مستطیل‌ها را محاسبه می‌کند.[۳۴]



شکل۳-۲ : مثال‌هایی از ویژگی‌های مستطیلی[۳۴]

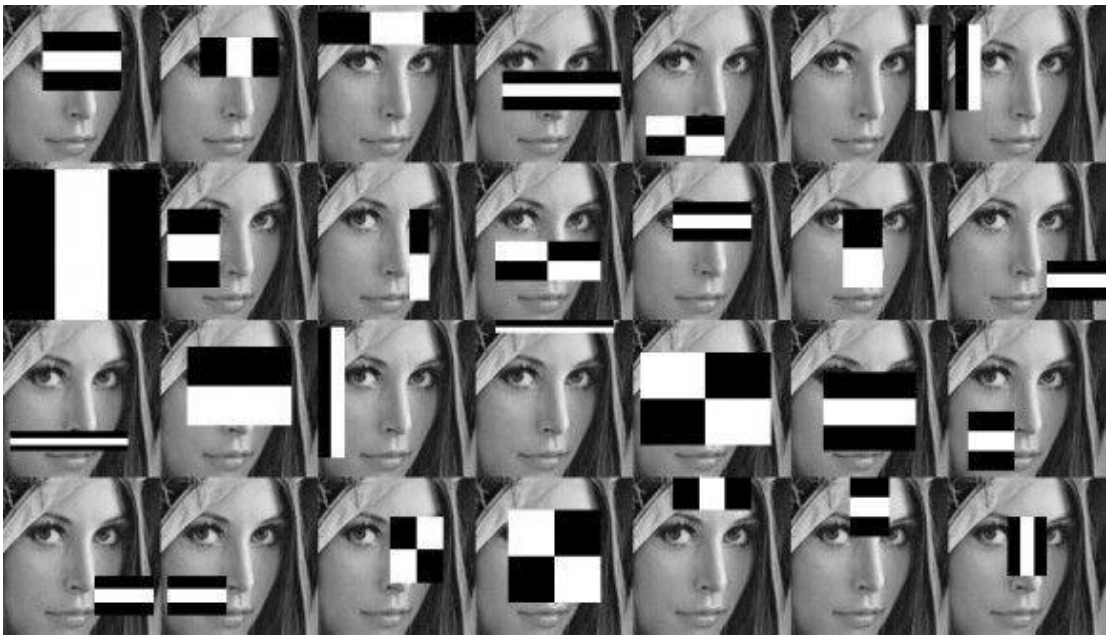
همانطور که در شکل۳-۲ مشاهده می‌کنید، مثال‌هایی از ویژگی‌های مستطیل که نسبت به پنجره تشخیص نمونه نشان داده است. مجموع پیکسل‌های که روی سطح مستطیل سفید قرار دارد از مجموع پیکسل در مستطیل خاکستری کسر می‌شود. ویژگی دو مستطیلی در (A) و (B) نشان‌داده‌شده-است. شکل (C) یک ویژگی سه مستطیلی را نشان می‌دهد و (D) یک ویژگی چهار مستطیلی است.[۳۴]

[1] Features Rectangle

۳-۴-۳- ویژگی‌های هار[1][۳۴]

ویژگی‌های هار-مانند ویژگی‌های تصویری دیجیتال هستند که در تشخیص شی مورد استفاده قرار می‌گیرند. آن‌ها نام خود را به واسطه ویولتس هارگرفته‌اند و در اولین شناسایی‌کننده چهره بلادرنگ به کار گرفته شدند. از لحاظ تاریخی، ابتدا این ویژگی‌ها تنها با شدت‌های تصویر کار می‌کردند. به عنوان مثال، مقادیر پیکسل قرمز، سبز و آبی در هر پیکسل تصویرکه این روش کار محاسباتی را گران و زمان-برکرده‌بود. یک مقاله توسط پاپاگورجیو و همکارانش در مورد کار با یک ویژگی جایگزین براساس داده‌های هار، به جای شدت‌های تصویر که معمولا مورد استفاده بود، منتشر شد. یک ویژگی هار- مانند در نظر گرفتن مناطق مستطیلی مجاور در یک مکان خاص در یک پنجره است، که برای تشخیص انتخاب شده‌است و محاسبه پیکسل‌ها در هر منطقه و تفاوت بین این مقادیر محاسبه شده است. سپس این تفاوت برای دسته‌بندی زیر بخش‌های یک تصویر استفاده می‌شود. برای مثال، برای شناسایی یک چهره انسانی، مشاهده علنی نشان‌دهنده این است که میان همه چهره‌ها، منطقه چشم‌ها تیره‌تر از ناحیه گونه‌ها است. از این رو، ویژگی هار برای تشخیص چهره، مجموعه‌ای از دو مستطیل مجاور است که در بالای چشم و ناحیه گونه قرار دارند.

[1] Haar features



شکل۳-۳- جایگیری ویژگی Haar[۳۴]

در شکل ۳-۳ مشخص است که این ویژگی‌ها محدود به شناسایی چشم و ناحیه‌ی گونه نیستند بلکه برای شناسایی جزئی‌ترین اعضای صورت کاربرد دارد. به گونه ایی که این مستطیل‌ها با ابعاد مختلف در راستای پنجره مورد نظر برای شناسایی چهره حرکت می‌کنند. حاصل محاسبات انجام شده مشخص شدن موقعیت چهره به صورت کادری مربع شکل است. این عملیات در حقیقت به چهار مرحله قابل تقسیم است که عبارتند از: محاسبه ویژگی‌های هار، ایجاد تصاویر انتگرالی، استفاده از آدابوست و اجرای کلاس‌بندی آبشاری

۳-۴-۳-۱-استفاده از آدابوست[۱][۳۵]

آدابوست یک تکنیک بوست[۲] محبوب است که به شما کمک می‌کند دسته‌بندی کننده‌های ضعیف چندگانه را در یک طبقه‌بندی کننده قوی ادغام کنید. طبقه‌بندی کننده ضعیف، یک طبقه‌بندی کننده ساده است که ضعیف عمل می‌کند, اما بهتر از حدس زدن تصادفی عمل می‌کند. یک مثال ساده برای این طبقه کننده این است که جنسیت یک فرد را براساس ارتفاع قد طبقه‌بندی کند.

[1] Adaboost

[2] Boost

شما می‌توانید بگویید که هر کسی بیش از یک متر و شصت سانتی‌متر باشد یک مرد و هر کس دیگری زیر این اندازه است، زن است. شما افراد زیادی را به این روش درست حدس نخواهید زد, اما دقت شما همچنان بزرگ‌تر از ۵۰ درصد خواهد بود.

آدابوست می‌تواند برای هر الگوریتم طبقه‌بندی اعمال شود، بنابراین این یک تکنیک است که بر روی مدلی که با طبقه‌بندی کننده خود طبقه‌بندی می‌شوند، ساخته می‌شود. شما می‌توانید یک دسته از طبقه‌بندی کننده‌های ضعیف را به تنهایی آموزش دهید و نتایج را با هم ترکیب کنید. اما به دو دلیل زیر این روش بهینه‌تر شدن سیستم را ضمانت می‌کند:

- این کار به شما کمک می‌کند یک مجموعه آموزشی را برای هر طبقه‌بندی کننده جدید انتخاب کنید که براساس نتایج طبقه‌بندی کننده قبلی آموزش داده شده است.
- مشخص می‌کند که چه مقدار وزن باید به پاسخ پیشنهادی طبقه بندی کننده در هنگام ترکیب نتایج، داده شود.

هر طبقه‌بندی کننده ضعیف باید بر روی یک زیرمجموعه تصادفی از مجموعه آموزشی کل، آموزش داده شود. این زیرمجموعه‌ها می‌توانند هم پوشانی داشته باشند. مثلاً به تقسیم کردن مجموعه آموزشی به ده بخش جدا از هم نیازی نیست. آدابوست یک وزن برای هر نمونه آموزشی تعیین می‌کند، که احتمال اینکه هر نمونه باید در مجموعه آموزشی ظاهر شود را مشخص می‌کند. نمونه‌های با وزن‌های بالاتر احتمال بیشتری دارند تا در مجموعه‌ی آموزشی گنجانده شوند و برعکس. بعد از آموزش طبقه‌بندی کننده‌های ضعیف، وزن روی نمونه‌های طبقه‌بندی نشده را افزایش می‌دهد به طوری که این نمونه‌ها بخش بزرگ‌تری از مجموعه آموزشی بعدی را تشکیل می‌دهند و خوشبختانه طبقه‌بندی کننده بعدی بررسی ویژگی آن‌ها را بهتر انجام خواهند داد.[۳۵]

۳-۴-۳-۲-کلاس‌بندی آبشاری[1]

در مرحله بعد همان‌طور که مطرح شده بود، این مدل تقویت‌شده به کلاس‌بندی آبشاری داده می‌شود. طبقه‌بندی کننده آبشاری از مجموعه‌ای از مراحل ساخته شده‌است که در آن هر مرحله

[1] Cascade classifier

مجموعه‌ای از یادگیرهای ضعیف است. یادگیرهای ضعیف با استفاده از بوست[1] آموزش داده می‌شوند. احتمال وجود ویژگی در پنجره مشخص شده پیش‌بینی می‌شود. با این پیش‌بینی، طبقه‌بندی کننده تصمیم می‌گیرند که نشان دهنده یک شی یافته شده‌است (مثبت) یا به سمت ناحیه بعدی حرکت می‌کند (منفی). [۳۵]

[1] Boost

فصل چهارم

معماری شبکه‌های عصبی

۴-۱- انواع شبکه های عصبی

همه این‌ها با لنت[1] در سال ۱۹۹۸ آغاز شد و در نهایت بعد از ۱۵ سال به مدل‌های شکسته شده منجر شد که شامل الکس‌نت[2] در سال ۲۰۱۲، گوگل‌نت[3] در سال ۲۰۱۵ و به رس‌نت[4] در سال ۲۰۱۵ و به ترکیبی از آن‌ها در سال ۲۰۱۶ شد. در ۲ سال گذشته، هیچ پیشرفت قابل‌توجهی صورت نگرفته است و مدل‌های جدید یک ترکیب از گروهی از مدل‌های شکسته‌شده‌ی قبلی هستند. شبکه‌های عصبی مصنوعی (سی ان ان) نوع خاصی از شبکه‌های عصبی چند لایه هستند که برای تشخیص الگوهای تصویری مستقیماً از پیکسل‌های تصاویر با حداقل پیش‌پردازش طراحی شده‌اند . پروژه امیجنت[5] یک پایگاه‌داده تصویری بزرگ است که برای استفاده در تحقیقات نرم‌افزاری تشخیص شی دیداری طراحی شده‌است . پروژه امیجنت یک رقابت نرم‌افزاری است که سالانه برگزار می‌گردد که در آن برنامه‌های نرم‌افزاری برای طبقه‌بندی صحیح اشیا و صحنه‌ها به رقابت می‌پردازند .این پروژه سنجش نسبتا مناسبی برای کاربرد و عملکرد این معماری‌های شبکه عصبی است. معماری‌هایی که توانسته‌اند در این سنجش‌ها موفقیت‌آمیز عمل کنند به ترتیب سال انتشار آن‌ها عبارتند از:

۱. لنت-۵ (LeNet-5)
۲. الکس‌نت(AlexNet)
۳. VGG-16
۴. اینسپشن-۱(Inception-v1)
۵. اینسپشن-۳(Inception-v3)
۶. رس‌نت-۵۰(ResNet-50)

[1] LeNet

[2] AlexNet

[3] GoogleNet

[4] ResNet

[5] ImageNet

۷. اکسپشن(Xception)

۸. انسپشن-۴(Inception-v4)

۹. اینسپشن-رس‌نت(Inception-ResNets)

۱۰. رس‌نکست-۵۰(ResNeXt-50)

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
VGG16	528 MB	0.713	0.901	138,357,544	23
InceptionV3	92 MB	0.779	0.937	23,851,784	159
ResNet50	98 MB	0.749	0.921	25,636,712	-
Xception	88 MB	0.790	0.945	22,910,480	126
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
ResNeXt50	96 MB	0.777	0.938	25,097,128	-

شکل ۴-۱- عملکرد تعدادی از معماری‌های شبکه عصبی در پروژه ایمج‌نت [۴۳]

در شکل ۴-۱ عملکرد شش عدد از معماری‌ها برای دو معیار و تعداد پارامترها و عمق شبکه قابل مقایسه است.

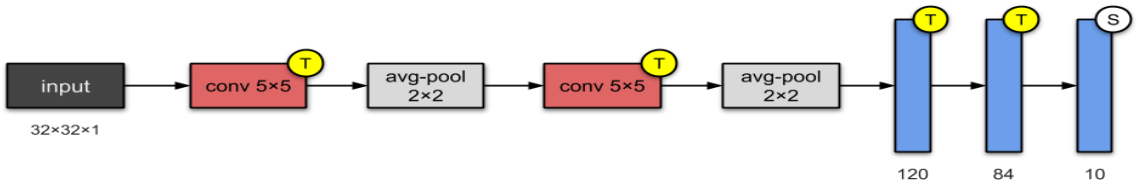
۴-۱-۱- لنت-۵[1]

این شبکه همچنین به عنوان شبکه عصبی کلاسیک شناخته می‌شود که توسط یان لکون، لئون بوتو، یوشا بنگیو و پتریک هفنر برای دست‌نوشته و تشخیص نوشته چاپی در دهه ۱۹۹۰ طراحی شده‌است که آن‌ها را لنت-۵ نامیدند. این معماری برای شناسایی ارقام دست‌نوشته شده در مجموعه دیتاست MNIST طراحی شده‌بود. معماری بسیار ساده و مستقیم است تا درک شود. تصاویر ورودی با ابعاد ۳۲ * ۳۲ * ۱ با دو جفت لایه کانولوشن در ۲ گام و پولینگ[2] متوسط‌گیر در یک گام دنبال شدند. در نهایت، لایه‌های کاملاً متصل شده با فعال‌سازی سافت‌مکس در لایه خروجی تکمیل شده اند. این شبکه ۶۰،۰۰۰ پارامتر در کل داشت.

[1] Lenet-5

[2] Average Pooling

از ویژگی‌های این معماری می توان به این که هر لایه کانولوشن شامل سه بخش است: کانولوشن, پولینگ و توابع فعال‌ساز غیر خطی اشاره‌کرد. استفاده از کانولوشن برای استخراج ویژگی‌های فضایی است. استفاده از MLP به عنوان آخرین طبقه‌بندی کننده و اتصال کامل بین لایه‌ها به منظور کاهش پیچیدگی محاسبات از دیگر ویژگی‌های این معماری است.[۳۶]



شکل ۴-۲- معماری شبکه عصبی لنت-۵[۴۴]

در شکل ۴-۲ قادر به مشاهده لایه‌های تشکیل‌هنده‌ی معماری شبکه‌س لنت ۵ است.

۴-۱-۲- الکس‌نت[1]

این شبکه بسیار شبیه لنت-۵ بود, اما با ۸ لایه, با فیلترهای بیشتری, لایه‌های کانولوشنی بیشتر, پولینگ حداکثر[2], دراپ‌اوت[3], افزایش داده‌ها, رلو و SGD بود. الکس‌نت برنده رقابت‌های ImageNet ILSVRC -2012 است که توسط الکس کرلزوسکی، الیا استاسکور و جفری ای طراحی شده‌است. این شبکه در دو جی‌فورس (580 gpus) آموزش‌دیده بود, بنابراین شبکه به دو خط مجموعه جدا تقسیم شد. الکس‌نت دارای ۵ لایه کانولوشنی و ۳ لایه کاملاً متصل است. عیب اصلی این شبکه این است که این شبکه متشکل از تعداد بیش از حد هایپر-پارامتر است. یک مفهوم جدید از نرمالیزیشن محلی[4] نیز در این مقاله مطرح شد. این معماری اولین بار از دارپ‌اوت و رلو استفاده کرد.

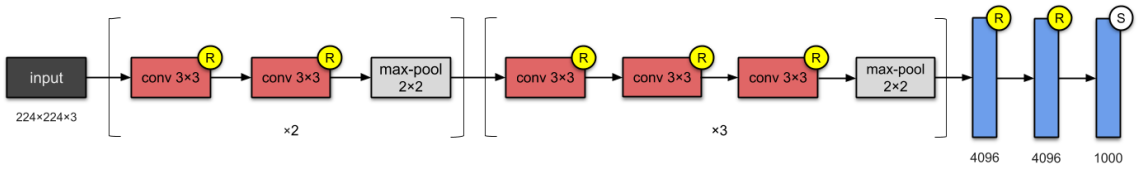
[1] AlexNet

[2] Max Pooling

[3] Dropout

[4] Local Normalization

الکس‌نت یکی از تاثیرگذارترین مقالات منتشر شده در بینایی رایانه‌ای است و بسیاری از مقالات منتشر شده توسط cnns و gpus را تحت تاثیر می‌گذارد تا یادگیری عمیق را تسریع کنند. تا سال ۲۰۲۱، به گفته گوگل، این مقاله به بیش از ۸۰ هزار بار دیده شده‌است.[۳۶]



شکل ۴-۳ معماری شبکهی عصبی الکس‌نت[۴۴]

در شکل ۴-۳ معماری دو قسمتی و لایه‌های تشکیل دهنده‌ی دو بخش را به صورت شماتیک ترسیم کرده‌اند.

VGG-16 -۴-۱-۳

VGG16 یک معماری شبکه عصبی کانولوشنی (CNN) است که برای برد رقابت ILSVR(امیجنت) در سال ۲۰۱۴ مورد استفاده قرار گرفت. این معماری به عنوان یکی از بهترین معماری ها تا به امروز در نظر گرفته می‌شود. نقطه ضعف اصلی الکس‌نت تعداد بیش از حد از پارامترهای بود که توسط VGG با جایگزین کردن فیلترهای با اندازه هسته‌ی بزرگ (به ترتیب ۱۱ و ۵ در اولین و دومین لایه کانولوشن) با چندین فیلتر با اندازه کرنل ۳ * ۳ یکی بعد از دیگری حل شد.

این معماری که توسط سیمونیون و زیسرمن توسعه یافت، اولین برنده از چالش تشخیص تصویری[۱] ۲۰۱۴ بود. معماری شامل فیلترهای ۳ * ۳ کانولوشنی ، پولینگ حداکثری[۲] با یک گام است. در مجموع ۱۶ لایه در شبکه وجود دارد که در آن تصویر ورودی با ابعاد ۲۲۴ * ۲۲۴ * ۳ و ۵ جفت از فیلتر کانولوشنی با اندازه‌های: (۶۴، ۱۲۸، ۲۵۶،۵۱۲،۵۱۲) و پولینگ حداکثری وجود دارد. خروجی این لایه‌ها به سه لایه کاملاً متصل داده شدند و یک تابع سافت‌مکس در لایه خروجی قرار گرفته می‌شود. در مجموع

[1] Visual Recognition Challenge

[2] Max Pooling

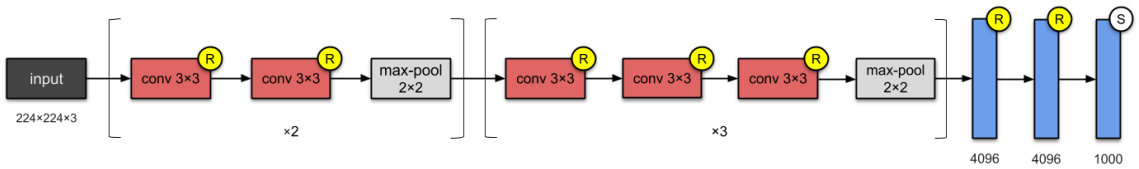
۱۳۸ میلیون پارامتر در VGG وجود دارد. با این حال این معماری دارای سه مشکل عمده بود که عبارتند از :

۱. آموزش مدل بسیار زمان‌بر بود.

۲. معماری به شت پرحجم و سنگین بود.

۳. از نظر محاسباتی گران شمرده می‌شد.

این معماری حدود پانصد مگابایت حافظه نیاز داشت که برای یک شبکه عصبی زیاد تلقی می‌شد.در مقاله منتشر شده معماری دیگری نیز بیان شد که VGG-19 نام‌گذاری شده‌بود. این معماری پیچیده-تر و حجیم‌تر بود. با توجه به دلایل ذکر شده، مورد توجه قرار نگرفت.[۳۷]



شکل ۴-۴ معماری شبکه‌ی عصبی VGG-16[۴۴]

در شکل ۴-۴ بخش های مختلف لایه‌های تشکیل دهنده‌ی معماری شبکه VGG-16 به صورت شماتیک ترسیم شده‌است.

۴-۱-۴-اینسپشن-۱(گوگل‌نت)

این معماری در سال ۲۰۱۴ موفق به کسب مقام در پروژه ایمجنت بود. که عموما به نام گوگل‌نت[1] معروف شد. این معماری ۲۲ لایه با ۵ میلیون پارامتر به نام اینسپشن-۱ نیز نامیده می‌شود. این کار با استفاده از

[1] GoogleNet

ماژول‌های اینسپشن انجام می‌شود. طراحی معماری مدل انسپشن محصول تحقیق در باب ساختارهای تقریبا پراکنده است. هر ماژول, ۳ ویژگی را در برمی‌گیرد:

۱. با استفاده از مجموعه‌ای موازی از کانولوشن‌ها با فیلترهای مختلف دسته‌بندی انجام می‌شود . این ایده توسط آرورا و همکارانش در مقاله پروبل[1] برای یادگیری برخی نمایش‌های عمیق ایجاد شد, که یک لایه - با ساختاری که در آن باید آمار همبستگی[2] لایه آخر را تجزیه و تحلیل کند و آن‌ها را به گروه‌هایی از واحدها با همبستگی بالا دسته‌بندی کند.

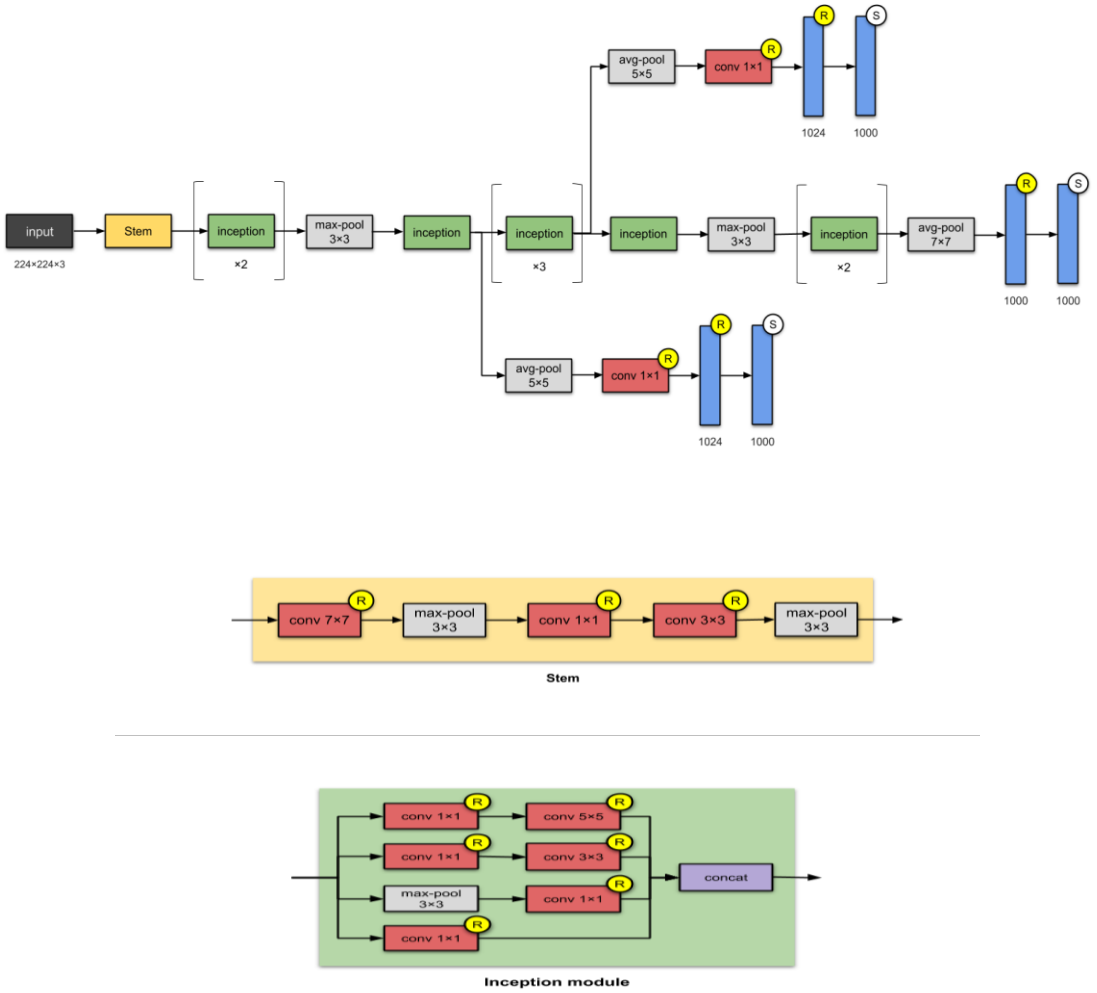
۲. کانولوشن‌های ۱*۱ برای کاهش ابعاد و در نتیجه کاهش سختی و پیچیدگی‌های محاسباتی استفاده شد.

۳. دو طبقه‌بندی کننده‌های کمکی را برای تبعیض در مراحل پایین‌تر طبقه بندی‌کننده, به منظور افزایش سیگنال گرادیان که به عقب تکثیر شده‌است و ایجاد رگولاریزیشن اضافی معرفی کردند. شبکه‌های کمکی (شاخه‌هایی که به طبقه‌بندی کننده کمکی متصل شده‌اند) در زمان استنتاج دور ریخته می‌شوند.

نوآوری که در این معماری نمایان شد استفاده از ماژول هایی از لایه‌های کانولوشنی به‌جای استفاده مستقیم از کانولوشنی بود. استفاده بهتر و بهینه‌تر از منابع رایانه‌ای در داخل مدل از محاسن های این معماری بشمار می‌رود.[۳۸]

[1] Provable

[2] Correlation statistics



شکل۴-۵- معماری شبکه‌ی عصبی اینسپشن-۱[۴۴]

در شکل ۴-۵ معماری کلی اینسپشن-۱ در قسمت فوقانی تصویر قابل مشاهده است که قسمت های زرد و سبز که به اختصار در مستطیل ها تعریف شده‌اند، در دو تصویر جدا بسط داده شده‌اند.

۴-۱-۵- اینسپشن-۳

اینسپشن-۳ یک شبکه عصبی مصنوعی برای کمک به آنالیز تصویر و تشخیص جسم است و به عنوان یک ماژول برای گوگل‌نت شروع شد. این سومین نسخه از شبکه عصبی کانولوشنی ماژول گوگل است که در ابتدا در طول چالش شناسایی ایمجنت معرفی شد. همانطور که ایمیچنت را می توان به عنوان یک پایگاه‌داده از اشیا بصری معرفی کرد، این معماری به طبقه‌بندی اشیا در دنیای کامپیوتری کمک

می‌کند. یکی از این کاربردهای اصلی آن در علوم زیستی است که در آن به تحقیقات در شناسایی لوکیمیا[1] کمک می‌کند.

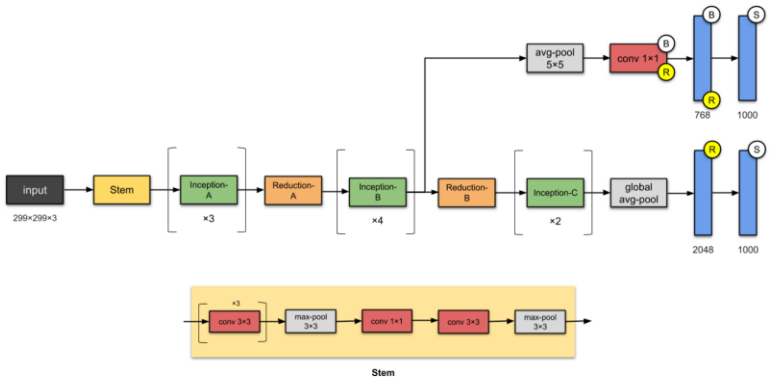
اینسپشن-۳ یک شبکه است که برای بهبود پیچش‌هایی مانند: اختلال در اپتمایزرها[2] یا بهینه-سازها, تابع زیان و افزودن نرمال سازی دسته‌ای[3] به لایه‌های کمکی در شبکه کمکی استفاده می‌شود. انگیزه برای این نسخه از اینسپشن این است که از تنگناها و سختی‌های اجرایی اجتناب کنیم (این به معنی کاهش قابل‌ملاحظه ابعاد ورودی لایه بعدی) و ارائه محاسبات کارآمد تری با استفاده از روش‌های فاکتورگیری است.[۳۹]

در مقایسه با نسخه اولیه نکات زیر در بهبود عملکرد این معماری کمک کرد:

۱. کانولوشن‌های نامتقارن n*n : به ترکیبی از ۱* n و ۱ * n تبدیل شدند.

۲. کانولوشن ۵ *۵ تبدیل به دو عملیات کانولوشن ۳ * ۳

۳. تعویض کانولوشن‌های ۷ * ۷ به یک سری از کانولوشن‌های ۳*۳

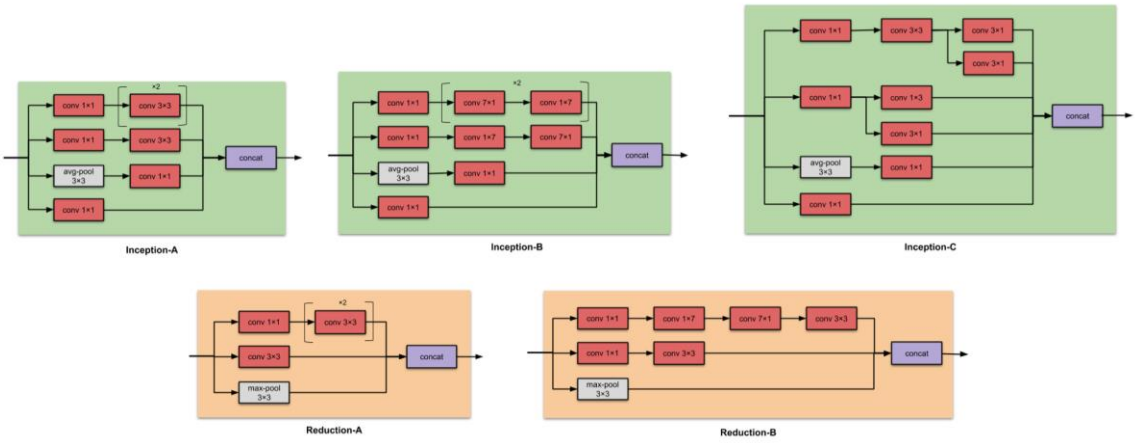


شکل ۴-۶- معماری شبکه‌ی عصبی نسخه سوم اینسپشن[٤٤]

[1] Leukemia

[2] Optimizer

[3] Batch normalization



شکل ۴-۷- معماری زیر ساخت‌های نسخه سوم اینسپشن[۴۴]

در شکل ۴-۶ معماری کلی اینسپشن-۳ قابل رویت است با زیر ساخت‌هایی که در شکل ۴-۷ به طور کامل به صورت شماتیک شرح داده شده‌است.

۴-۱-۶- رس‌نت[1]

رس‌نت در سال ۲۰۱۵ قهرمان رقابت‌های ایمیج‌نت بود و میزان خطا در طبقه‌بندی تصویر را ۳/۶٪ کاهش داد. این نتیجه حتی از دقت تشخیص چشم انسان معمولی نیز فراتر رفت. از طریق یادگیری مدل‌های کلاسیک قبلی، می‌توانیم دریابیم که با توسعه مستمر یادگیری عمیق، تعداد لایه‌ها مدل رو به افزایش است، ساختار شبکه پیچیده‌تر و پیچیده‌تر می‌شود. در تئوری، با فرض اینکه لایه‌های جدید اضافه‌شده همه در خدمت تشخیص مپینگ‌ها[2] هستند، تا زمانی که لایه‌های اصلی پارامترهای مشابه‌ای را به مدل اصلی یاد بدهند، سپس ساختار لایه‌های عمیق می‌توانند به تاثیر ساختار مدل اصلی دست یابد. هرچه این تاثیرگذاری عمیق‌تر شود دقت مدل کلی بهتر می‌شود. اما در عمل طبق نظریه‌های تئوری پیش نرفت. با این حال، در عمل نشان می‌دهد که پس از افزایش تعداد مجموعه لایه‌های شبکه، خطای آموزشی

[1] ResNet

[2] mappings

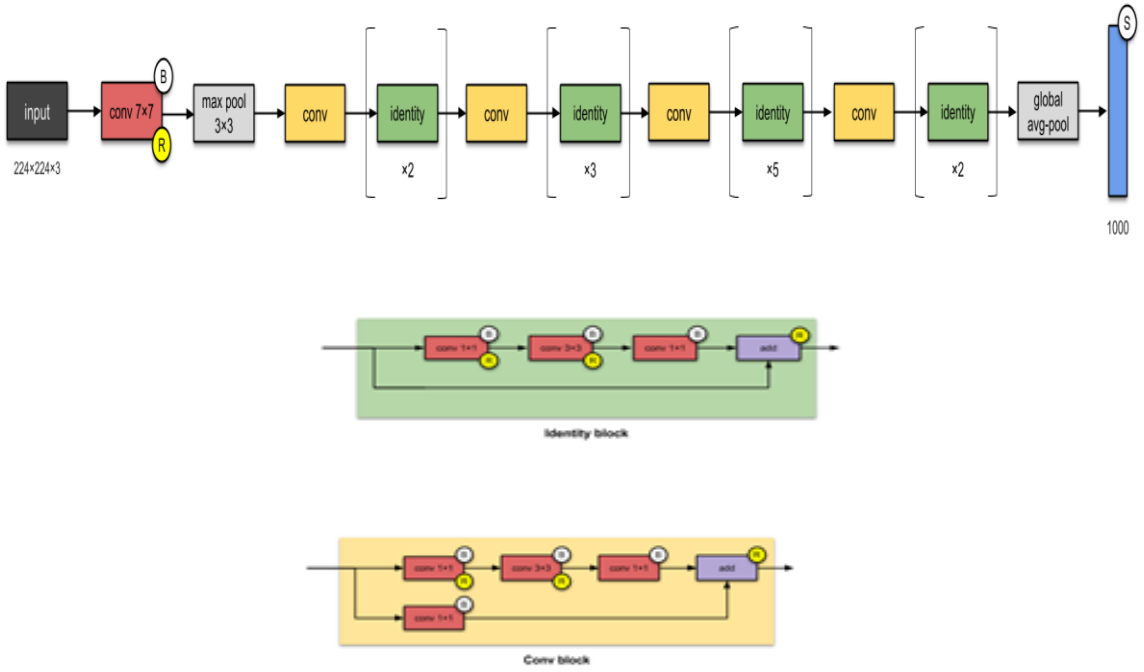
به جای کاهش، افزایش می‌یابد. پس در این راستا سعی بر یک سوسازی معماری طراحی شده شد.این نکته رس‌نت را پدید آورد.

رس‌نت یک شبکه عصبی مصنوعی است که در عمق ۵۰ لایه قرار دارد. شما می‌توانید یک نسخه از پیش آموزش دیده را از اینترنت دریافت کنید که در آن بیش از یک میلیون تصویر از پایگاه‌داده ایمیج‌نت آموزش‌دیده اند. شبکه از پیش آموزش داده می‌تواند تصاویر را به ۱۰۰۰ دسته شئی مانند صفحه‌کلید, مویس, مداد و بسیاری از حیوانات طبقه‌بندی کند. در نتیجه, این شبکه ویژگی‌های زیادی برای طیف وسیعی از تصاویر را فرا گرفته‌است. این شبکه تصویر با اندازه ورودی ۲۲۴ * ۲۲۴ را به عنوان ورودی دریافت می‌کند.[۴۰]

از دیگر ویژگی‌های منحصر به فرد این معماری می توان به :

- معروف کردن استفاده از رد کردن اتصال[1].
- طراحی حتی عمیق‌تر (تا ۱۵۲ لایه) بدون این که توان تعمیم مدل را به خطر اندازد.
- در میان اولین استفاده کننده‌ها از نرمال سازی دسته‌ای.

[1] Skip connection



شکل ۴-۸ معماری شبکه‌ی عصبی رس‌نت[۴۴]

در شکل ۴-۸ شماتیک معماری رس‌نت در قسمت فوقانی با بسط دو زیر ساخت به رنگ‌های سبز و زرد در زیر آن قاب مشاهده است.

۴-۱-۷-اکسپشن [1]

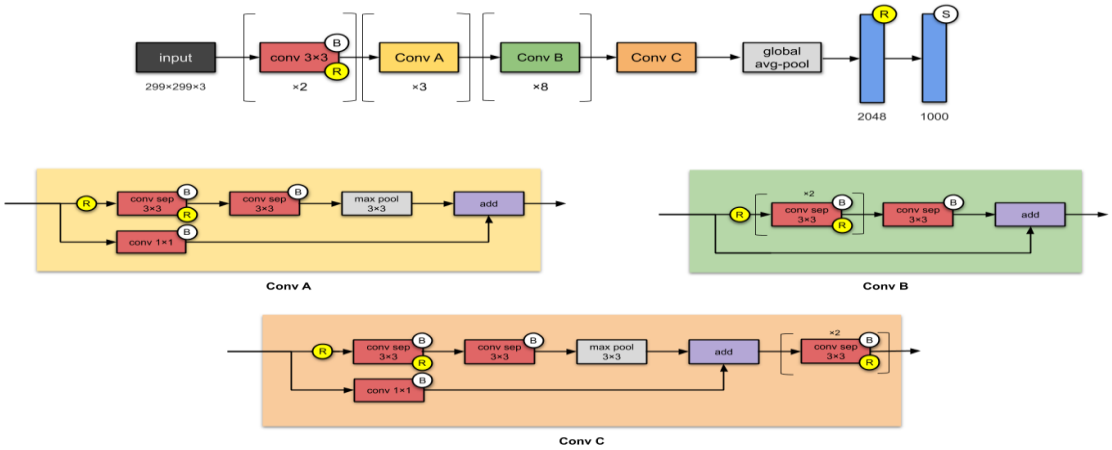
اکسپشن یک اقتباس از اینسپشن است که در آن واحدهای اینسپشن با کانولوشن قابل‌تفکیک در عمق جایگزین شده‌اند. همچنین در تعداد پارامتر همان تعداد از پارامترهای اینسپشن را دارد.(۲۳ میلیون) معماری اکسپشن ۳۶ لایه کانولوشنی دارد که پایه استخراج ویژگی را تشکیل می‌دهند .از شبکه برای بررسی و طبقه‌بندی تصویر استفاده می‌شود، در نتیجه در مدل پایه کانولوشن‌ها با یک رگرسیون منطقی

[1] Xception

دنبال خواهند شد. ۳۶ لایه کانولوشنی به شکلی زیر ساخت یافته‌اند که چهارده ماژول تشکیل می‌دهند که همه آن‌ها، به جز اولین و آخرین دارای اتصالات خطی هستند.

داده‌ها ابتدا از قسمت ورودی عبور می‌کنند, سپس از طریق قسمت میانی که هشت بار تکرار می‌شود به قسمت انتهای ساختار می‌رسند. توجه داشته باشید که همه لایه‌های کانولوشنی و کانولوشن‌های تفکیک‌پذیر توسط نرمال سازی دسته‌ای به خروجی می‌رسند.

ابتدا, همبستگی بین کانال (یا نقشه ویژگی‌های متقاطع) با کانولوشن‌های ۱ * ۱ بررسی می‌شوند. در نتیجه، همبستگی فضایی درون هر کانال از طریق کانولوشن‌های ۳ * ۳ یا ۵ * ۵ شکل می‌گیرد. با در نظر گرفتن این ایده به معنی انجام کانولوشن ۱ * ۱ برای هر کانال, سپس اجرای کانولوشن ۳ * ۳ برای هر خروجی در این معماری به معنی جایگزین کردن مدل اینسپشن با کانولوشن‌های قابل‌تفکیک در عمق است. [۴۱]



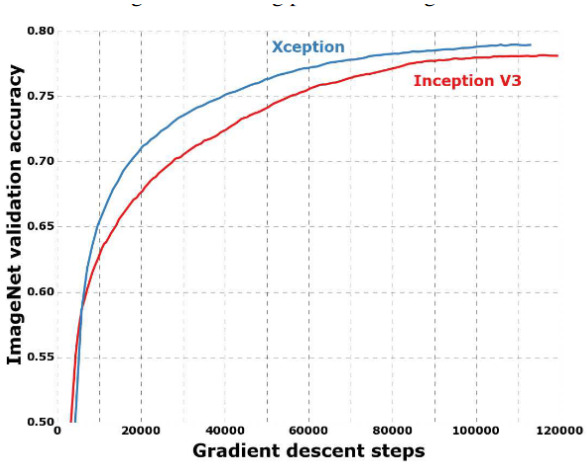
شکل۴-۹- معماری شبکه عصبی اکسپشن[۴۴]

در شکل ۴-۹ معماری شبکه‌ی عصبی اکسپشن به همراه سه زیر ساخت نام‌گذاری شده در سه رنگ نارنجی، زرد و سبز به نمایش گذاشته شده‌است.

۴-۱-۷-۱- مقایسه در دقت

معماری مطرح شده به دلیل شباهت فرایند کلی، مورد قیاس با هم‌خانواده‌های خود قرار گرفته است . تحقیقات انجام شده بیانگر دقت و سرعت این معماری نسبت به معماری های پیشین هستند.برای

مثال در دقت این معماری در آموزش دیتاست امیجنت به دقت معادل ۰/۷۹ همگرا می‌شود با اینکه معماری نسخه سوم اینسپشن به دقت در حدود ۷۸۲/. همگرا می‌شود. در این آزمایش دو معماری VGG16 و رس‌نت نیز دقت هایی معادل ۷۱۵/. و ۷۷/. بدست آورده‌اند.



شکل ۴-۱۰- مقایسه دقت دو معماری اکسپشن و اینسپشن-۳[۴۴]

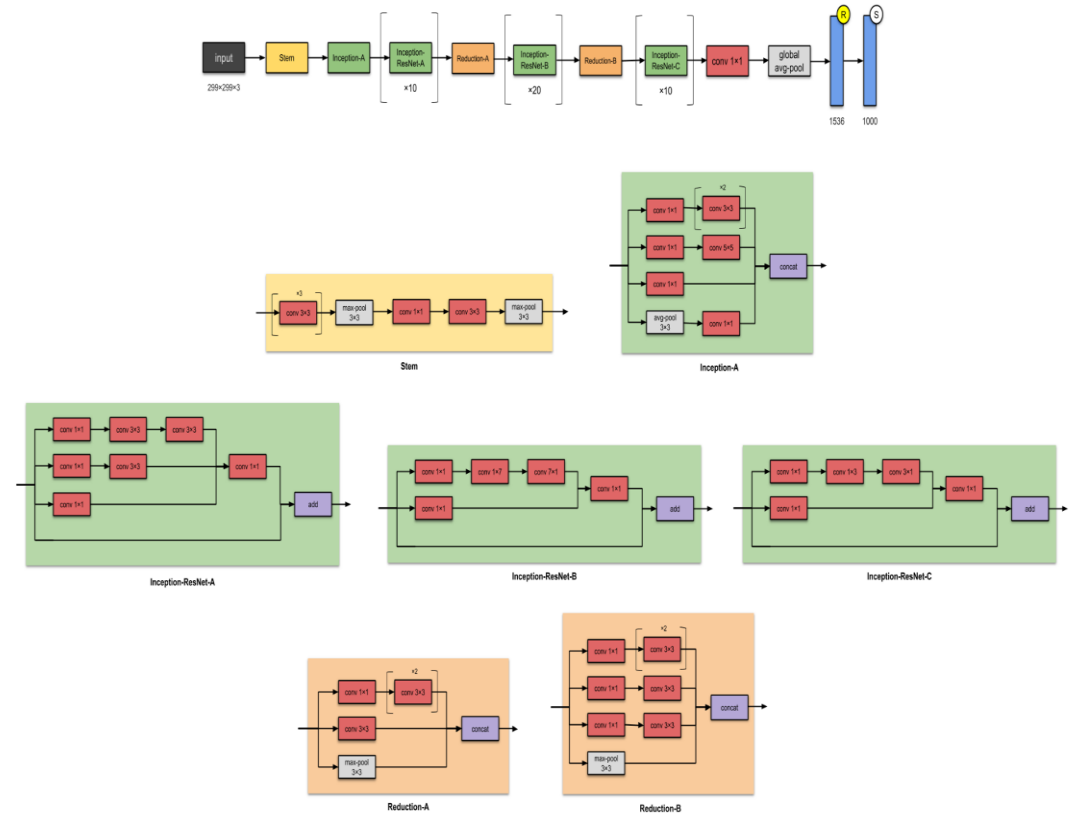
در شکل ۴-۱۰ به مقایسه دقت دو معماری اکسپشن و اینسپشن-۳ در استاندارد امیجنت پرداخته شده‌است. در محور عمودی دقت و در محور افقی تعداد مراحل گرادیان گسسته نمایش داده شده-است.[۴۱]

۴-۱-۸- اینسپشن-رس‌نت

در این معماری ،همانطور که از نام‌گذاری مشخص است ترکیبی از دو معماری اینسپشن و رس‌نت را مورد مطالعه قرار می‌دهیم. در این مدل ترکیبب از ایده‌های جدیدی حاصل شد. اتصالات باقیمانده توسط کروزوسکی و همکارانش و آخرین نسخه تصحیح شده از معماری اینسپشن در آن با هم ترکیب شدند.در این معماری اتصالات باقیمانده از اهمیت ذاتی برای آموزش ساختارهای بسیار عمیق برخوردار است .

از آنجا که شبکه‌های اینسپشن تمایل دارند بسیار عمیق باشند ، جای‌گذاری سکوی الحاق فیلتر ها در معماری اینسپشن با اتصالات باقی مانده امری طبیعی است. این کار به اینسپشن اجازه می‌دهد تا تمامی مزایای ویژگی‌های باقیمانده را در حالی که کارایی محاسباتی آن را حفظ می‌کند، به دست آورند.

علاوه بر به کارگیری انتگرال مستقیم، اینسپشن خودش می‌تواند با عمیق‌تر و گسترده‌تر کردن خود کارایی بیشتری داشته باشد. برای این منظور، یک نسخه جدید به نام اینسپشن-رس‌نت طراحی شده‌است، که یک شکل از معماری ساده‌تر و با ماژول‌های اولیه را در مقایسه با اینسپشن-۳ دارد[۴۱].



شکل ۴-۱۱ معماری شبکه‌ی عصبی اینسپشن-رس‌نت[۴۴]

در شکل ۴-۱۱ معماری شماتیک شبکه‌ی عصبی اینسپشن-رس‌نت به همراه هفت زیر ساخت نام‌گذاری شده ترسیم شده‌است.

فصل پنجم

تعاریف در پیاده‌سازی و شبیه‌سازی

۵-۱- بیان مسئله

ورودی تصویر که می‌تواند به صورت بیدرنگ از دوربین دریافت شود یا فایلی ذخیره شده باشد ابتدا وارد بلاک تشخیص صورت می‌شود در اینجا در صورت تطبیق با شرایط هار-کسکید[1] وجود یک یا چند صورت انسان یا عدم وجود آن تشخیص داده می‌شود سپس این پنجره به منظور استفاده به عنوان ورودی شبکه عصبی کانولوشنالی، پیش پردازش بر روی آن اعمال می‌شود که نهایتا تبدیل به یک تصویر حاوی صورت به ابعاد ۶۴×۶۴ پیکسل می‌گردد، پیش پردازش بدین منظور اعمال می‌شود تا تمامی ورودی ها به شبکه عصبی دارای ویژگیهای یکسانی همچون ابعاد تصویر گردند. سپس تصویر حاوی صورت به عنوان ورودی به شبکه عصبی کانولوشنالی داده می‌شود در این قسمت پس از اعمال لایه های کانولوشن، پولینگ و لایه‌های متصل منطبق بر معماری انتخاب شده برای شبکه عصبی کانولوشنال یکی از هفت حالت چهره به عنوان خروجی شبکه استخراج می‌گردد.

۵-۲- پردازش تصویر

اگر در ساده‌ترین حالت ممکن به تعریف یک تصویر بپردازیم می‌توانیم یک تصویر سیاه و سفید را مجموعه‌ای از یک آرایه یا ماتریس دو بعدی تعریف کرد. درایه‌های آن نشان دهنده‌ی پیکسل‌ها هستند. درایه‌های ماتریس یا آرایه مقادیری بین ۰ تا ۲۵۵ را می‌گیرند. صفر در حقیقت معدل رنگ سیاه در تصویر است و رنگ سفید با ۲۵۵ نشان داده می‌شود. طبق تعریف هرچه این مقدار به صفر نزدیک باشند، پیکسل تیره‌تر است. هرچه این درایه به ۲۵۵ نزدیک‌تر باشد پیکسل دارای رنگ روشن‌تری است. در شبکه های عصبی معمولا تصاویر در دو حال کلی به عنوان ورودی داده می‌شوند. این دوحالت سطح خاکستری[2] و تصویر رنگی[3] هستند.

---

[1] Haar-Cascade

[2] Gray scale image

[3] RGB image

۵-۲-۱- تصویر خاکستری در پردازش

تصویر خاکستری سطحی‌ترین و ساده‌ترین نوع تصویر برای پردازش است زیرا در یک ماتریس دو بعدی گنجانده می‌شود.

در طرفی دیگر، تصاویر رنگی ساختار متفاوتی نسبت به تصاویر خاکستری دارند. این تصاویر از سه سطح متفاوت که شامل صفحه‌ی قرمز(R)، صفحه‌ی سبز(G) و صفحه‌ی آبی(B) است. مقادیر هر درایه از صفحه مانند یک صفحه خاکستری مقادیری بین صفر تا ۲۵۵ را با توجه به سصح روشنایی در رنگ خود انتخاب می‌کنند. از ترکیب این سه رنگ رنگ نهایی حاصل می‌شود. در هدف این پروژه که در کل شناسایی احساسات در قالب چهره است رنگ‌ها فاکتور بسیار کم تاثیری در طبقه‌بندی شبکه محسوب می‌شوند. رنگ پوست فرد یا آرایش در چهره و حتی رنگ چشم افراد در بیان احساسات آن ها نقشی ندارد بلکه جای‌گیری عضلات چهره به دور از تاثیر نژاد و قومیت به طبقه‌بندی این مسئله کمک می‌کند.

در انتخاب دیتاست نیز می‌توان به دلیل‌های ذکر شده در بالا و از همه مهم‌تر برای سادگی و جلوگیری از پردازش‌های پیچیده در فراهم‌سازی مدل از دیتاست‌های با تصاویر خاکستری استفاده کرد. در مورد دیتاست FER که در پروژه استفاده شده است این نکته که تصاویر آموزش در سطح خاکستری هستند ذکر شده‌است.[۴۵]

شکل ۵-۱- تصویر رنگی و سه صفحه قرمز[۴۵]

در شکل ۵-۱ به تلفیق سه طیف رنگی متشکل از قرمز، سبز و آبی برای پدید آوردن تصویر رنگی در سمت راست پرداخته شده‌است.

۵-۲-۲- آستانه‌گذاری در تصاویر[1][۴۵]

در پردازش دیجیتال تصاویر، ترش‌هلدینگ از ساده‌ترین روش‌های قطعه قطعه‌سازی[2] تصاویر محسوب می‌شود. در کنار این روش، روش‌های خوشه‌سازی مانند الگوریتم‌های k-mean نیز مورد استفاده قرار می‌گیرد. ما در این پروژه از ترش‌هلدینگ اتوماتیک استفاده کردیم.

آستانه گذاری خودکار راهی خوب برای استخراج اطلاعات مفید کد گذاری شده هر پیکسل از تصویر است در حالی که نویز پس‌زمینه را به حداقل می‌رساند. این کار با استفاده از یک حلقه فیدبک برای بهینه‌سازی ارزش آستانه‌ی هر پیکسل قبل از تبدیل تصویر سیاه و سفید به تصویر باینری انجام می‌شود. ایده این است که تصویر را به دو بخش پس‌زمینه و پیش‌زمینه تقسیم شود.

مراحل این آستانه‌گذاری به شرح زیر است:

۱. انتخاب آستانه اولیه که معمولا مقدار میانگین هشت بیتی تصویر اصلی است.

۲. تقسیم کردن تصویر به دو بخش کلی:

أ. مقدار پیکسل‌هایی که از مقدار آستانه کمتر یا مساوی آن هستند.(پس‌زمینه)

ب. مقدار پیکسل‌هایی که از مقدار آستانه بیشترا هستند.(پیش‌زمینه)

۳. محاسبه میانگین برای دو بخش جداشده.

۴. محاسبه آستانه جدید از میانگین دو مقدار محاسبه شده در مرحله قبل.

۵. این چهار مرحله تا زمانی که حدفاصل بین دو آستانه(قبلی و جدید) از مرزی کمتر شود ادامه پیدا می‌کند.

[1] Thresholding

[2] Segmentation

۵-۲-۳- آماده سازی دیتاست

در این بخش باید دیتاست را برای انجام عملیات مطرح‌شده و برای داده شدن به عنوان ورودی برای آموزش شبکه عصبی آماده کنیم. این مرحله ابتدا با تبدیل کردن آرایه به ماتریس دو بعدی، استاندارد تصویر شروع می شود. همانطور که در توضیح دیتاست FER قابل مطالعه است این دیتاست از دو آرایه که یکی دارای ۲۳۰۴ درایه با مقدار دهی بین صفر تا ۲۵۵ است و دیگری دارای یک درایه بین صفر تا ۶ است. ابتدا آرایه اول را به یک ماتریس ۴۸ در ۴۸ تبدیل می‌کنیم که طول و عرض تصاویر را تعریف می کند. سپس انجام آستانه‌گذاری برای راحتی پردازش و کاستن از بار محاسباتی تصاویر را به تصاویری باینری تبدیل می‌کنیم. به این تریتب که:

۱. مقادیر پیکسل‌ها را بر ۲۵۵ تقسیم می کنیم تا مقادیری بین صفر تا یک را بگیرند.

۲. از مقدار هر پیکسل مقدار نیم واحد را می‌کاهیم حال مقادیر بین منفی نیم تا مثبت نیم واحد هستند.

۳. با دو برابر کردن این مقادیر بازه پیکسل‌ها به [۱و۱-] تبیدل می‌شود و تصویر باینری است.

۵-۲-۴- دسته بندی تصاویر برای آموزش و تست

در آموزش شبکه‌ی عصبی مصنوعی در کل باید داده‌ها به دو دسته‌ی کاملا جدا تقسیم شود. این دو دسته داده‌های تست و آموزش[1] نامیده شدند. اکثرا دسته‌ی آموزش حداکثر داده‌ها را به خود اختصاص می‌دهد. این جداسازی در مفهوم اصلی خود برای محک دقت و صحت مدل آموزش دیده است به این صورت که ابتدا آموزش کلی شبکه توسط داده‌های آموزش انجام می‌شود با این حال که در این مرحله نیز می توان دقت را برای شبکه محاسبه کرد ولی این مقدار با توجه به این که داده‌ها دوباره به شبکه داده می‌شوند و قبلا شبکه دقیقا همین تصاویر را به عنوان ورودی دریافت کرده‌است، صلاحیت کافی ندارد. در مرحله تست داده‌های تست که برای شبکه جدید هستند به عنوان ورودی، داده می‌شوند و حال خروجی محک تقریبا بهتری برای ارزیابی دقت مدل است.

[1] Test and train

برای جداسازی دیتاست به تست و آموزش تعدادی روش ازجمله جداسازی تصادفی و جداسازی از روش k-mean وجود دارد. روش دوم در مواقعی که پراکنگی در تعداد داده‌ها دیده می‌شود مورد استفاده قرار می‌گیرد. در این پروژه سعی بر انجام هر دو روش برای جداسازی داده ها شده‌است. روش تصادفی با استفاده از کد train_test_split و روش دوم با استفاده از کد cross_val_score انجام شده‌است. هردو ماژول را می توانید در کتابخانه scikit learn پیدا کنید. در مقایسه عملکرد تفاوتی دیده نشد. در همین باب سعی بر استفاده از روش تصادفی شد زیرا هم سریع تر است و هم بار محاسباتی کمتری برای رایانه دارد و در نتیجه به صرفه‌تر است.

۵-۳- بهینه‌سازی[۴۶]

بهینه‌سازها[1] الگوریتم یا روش‌هایی هستند که برای تغییر ویژگی‌های شبکه عصبی مثل وزن‌ها و نرخ یادگیری برای کاهش تلفات مورد استفاده قرار می‌گیرند. بهینه‌سازها برای حل مسائل بهینه‌سازی با به حداقل رساندن تلفات، مورد استفاده قرار می‌گیرند.

بیهنه‌سازهای زیادی وجود دارند. هر یک از آن‌ها دارای مزایا و معایبی هستند که اغلب مربوط به وظیفه ویژه‌ای که در شبکه انجام می‌دهند، است. بهینه‌سازها را به دو خانواده تقسیم می‌شوند: بهینه‌ساز نزولی گرادیان و بهینه‌ساز تطبیقی. این بخش‌بندی منحصراً مبتنی بر جنبه عملیاتی است که شما را مجبور می‌کند تا به طور دستی نرخ یادگیری را در مورد الگوریتم نزولی گرادیان تنظیم کنید در حالی که این امر به طور خودکار در الگوریتم‌های تطبیقی انجام داده می‌شود. مثال‌هایی از این دو دسته در ادامه مطرح شده‌است.

نزول گرادیان :

۱. نزول گرادیان دسته‌ای
۲. نزول گرادیان تصادفی
۳. نزول گرادیان دسته‌ای کوچک

[1] Optimizers

تعداد الگوریتم‌های تطبیقی زیاد است تعدادی از معروف‌ترین ها عبارتند از:آداگارد، آدادلتا، آدام و ...

۵-۳-۱- بهینه‌ساز نزولی گرادیان[1]

نزول گرادیان ابتدایی‌ترین و ساده‌ترین الگوریتم بهینه‌سازی است. این روش به شدت در رگرسیون خطی و الگوریتم‌های طبقه‌بندی استفاده می‌شود. بک‌پروپوگیشن[2] در شبکه‌های عصبی نیز از الگوریتم گرادیان نزولی استفاده می‌کند. در ادامه به شرح سه دسته از این خانواده می‌پردازیم.

۵-۳-۱-۱- نزول گرادیان دسته‌ای[3]

با نام گرادیان وانیل شناخته می‌شود, این مهم‌ترین الگوریتم بین این سه دسته است. این مدل، شیب‌های تابع هدف J را با توجه به پارامترهای θ برای کل مجموعه آموزشی محاسبه می‌کند. به دلیل اینکه از کل مجموعه داده‌ها فقط در یک مرحله استفاده می‌کنیم, نزول گرادیان دسته‌ای می‌تواند بسیار کند عمل کند. علاوه بر این, برای مجموعه داده‌هایی که در حافظه، فضای کافی برای ذخیره‌سازی ندارد، مناسب نیست.

فرمول الگوریتم : $\theta=\theta-\alpha\cdot\nabla J(\theta)$

این الگوریتم ممکن است در حداقل محلی به دام بیفتد. وزن‌ها بعد از محاسبه گرادیان کل مجموعه داده‌ها تغییر می‌کنند. بنابراین، اگر مجموعه داده‌ها بسیار بزرگ باشد، ممکن است سال‌ها طول بکشد تا به حداقل برسد. به حافظه بزرگ نیاز دارد تا گرادیان کل مجموعه داده‌ها را محاسبه کند.

[1]Gradient Descent Optimizer

[2] Backpropagation

[3]Batch gradient descent

۵-۳-۱-۲- نزول گرادیان تصادفی[1]

این یک نسخه بهبود یافته از نزول گرادیان دسته‌ای است. به جای محاسبه گرادیان روی کل مجموعه داده‌ها, به روز رسانی پارامتر را برای هر نمونه در مجموعه داده‌ها انجام می‌دهد. بنابراین این فرمول در حال حاضر به مقادیر ورودی x و خروجی y بستگی دارد. مشکل این الگوریتم این است که به روزرسانی ها مکرر و با واریانس زیاد است, بنابراین تابع هدف به شدت در طول آموزش نوسان می‌کند. این نوسان می‌تواند یک مزیت با توجه به نزول گرادیان دسته‌ای باشد زیرا اجازه می‌دهد تا عملکرد به حداقل محلی سریع‌تر برسد، اما در عین حال می‌تواند یک عیب باشد با توجه به اینکه امکان عدم هم‌گرایی در حداقل محلی هست. یک راه‌حل برای این مشکل، کاهش تدریجی مقدار نرخ یادگیری به منظور به روزرسانی های کوچک‌تر،در راستای اجتناب از نوسانات زیاد است.

فرمول الگوریتم:

$\theta=\theta-\alpha\cdot\nabla J(\theta;x(i);y(i))$, where $\{x(i),y(i)\}$ are the training examples.}

مزیت‌ها :

به روزرسانی مکرر پارامترها منجر می‌شود در زمان کمتری همگرا شود.

به حافظه کمتری نیاز دارد چون نیازی به ذخیره مقادیر توابع زیان[2] ندارد.

معایب :

واریانس زیاد در پارامترهای مدل دیده می‌شود.

می توان حتی پس از رسیدن به حداقل جهانی، ناهمگرا شود.

برای رسیدن به همان هم‌گرایی، باید به آرامی مقدار نرخ یادگیری کاهش داده شود.

[1] Stochastic Gradient Descent

[2] Loss function

۵-۳-۱-۳- نزول گرادیان دسته‌ای کوچک

این روش در میان تمام انواع الگوریتم های نزول گرادیان، بسیار بهتر است. این یک بهبود بر روی SGD و نزول گرادیان استاندارد است(دو روش اول). این کار پارامترهای مدل را بعد از هر دسته به روزرسانی می‌کند. بنابراین مجموعه داده‌ها به دسته‌های مختلفی تقسیم می‌شوند و بعد از هر دسته, پارامترها به روز می‌شوند.

فرمول الگوریتم :

$\theta=\theta-\alpha\cdot\nabla J(\theta; B(i))$, where $\{B(i)\}$ are the batches of training examples.

مزیت‌ها:

پارامترهای مدل را در زمان‌بندی مناسبی به روز رسانی می‌کنند و واریانس کمتری نیز دارند.

به مقدار متوسطی از حافظه نیاز دارد.

۵-۳-۲- الگوریتم‌های تطبیقی[۴۶]

این خانواده از بهینه‌سازها برای حل مسائل الگوریتم‌های گرادیان نزولی معرفی شده‌است. مهم‌ترین ویژگی آن‌ها این است که آن‌ها نیاز به تنظیم مقدار نرخ یادگیری ندارند. در حقیقت برخی کتابخانه‌ها - مانند Keras - هنوز به شما اجازه می‌دهند که به طور دستی آن را برای آزمایش‌های پیشرفته تنظیم کنید. با وجود این حقیقت، هدف اصلی بر خودکار بودن این تنظیمات است.

در ادامه تعدادی از الگوریتم‌های این دسته شرح داده شده‌است.

۵-۳-۲-۱- آداگارد[1]

آداگارد الگوریتمی برای بهینه‌سازی مبتنی بر گرادیان است. این الگوریتم نرخ یادگیری را با پارامترهای کوچک‌تر تطبیق می‌دهد و به روزرسانی ها را برای پارامترهای مرتبط کوچک‌تر می‌کند(نرخ

[1] Adagard

یادگیری پایین). به روزرسانی های بزرگ‌تر (نرخ یادگیری بالا) برای پارامترهای مرتبط با ویژگی‌های نادر است.

به همین دلیل, برای پروژه‌هایی با داده‌های پراکنده مناسب است. دین و همکارانش دریافته اند که آداگارد به شدت مقاومت را نسبت به الگوریتم‌های قبلی بهبود بخشیده و از آن برای آموزش شبکه‌های عصبی در مقیاس بزرگ در گوگل استفاده کرده‌است و توانستند به شبکه آموزش دهند تا گربه‌ها را در ویدیوهای یوتیوب بشناسند.

در قانون به روز رسانی, آداگارد نرخ یادگیری عمومی η را در هر گام زمانی t برای هر پارامتر θi براساس گرادیان قبلی که برای θi محاسبه شده‌است, اصلاح می‌کند:

$$\theta_{t+1,i}=\theta_{t,i}-\eta/(\sqrt{(G_{t,ii}+\epsilon)})\cdot g_{t,i}.$$

مزیت‌ها:

یکی از مزایای اصلی آداگارد این است که نیاز به تنظیم دستی نرخ یادگیری را از بین می‌برد.

اکثر پیاده‌سازی‌ها از مقدار پیش‌فرض ۰/۰۱ استفاده می‌کنند.

معایب:

تجمع گرادیان مربع در مخرج مشکل اصلی این الگوریتم است. از آنجایی که هر عبارت اضافه‌شده مثبت است، مجموع در طول آموزش در حال افزایش است. این به نوبه خود باعث کاهش نرخ یادگیری می‌شود و در نهایت آنقدر کوچک می‌شود که الگوریتم دیگر قادر به کسب اطلاعات اضافی نیست.

۵-۳-۲-۲- آدادلتا[1]

این یک بسط از آداگارد است که باعث حذف مشکل کاهش سرعت یادگیری آن می‌شود. به جای جمع کردن تمام شیب‌های مربع پیشین، آدادلتا محدوده‌ای برای تعداد گرادیان‌های پیشین تعیین می‌کند. در این حالت مخرج را از بزرگ شدن با مجموع تمام گرادیان‌های مورد استفاده وامی‌دارد و به جای آن از جمع تعدادی مشخص از مربعات گرادیان استفاده می‌کند.

$$E\,[g^2](t)=\gamma.E[g^2](t-1)+(1-\gamma).g^2(t)$$

مزیت:

اکنون نرخ یادگیری رو به زوال نیست و آموزش متوقف نمی‌شود.

معایب:

از نظر محاسباتی پیچیده است.

۵-۳-۲-۳- آدام[2]

آدام با مومنتوم‌ها [3]از مرتبه اول و دوم کار می‌کند. شهود پشت سر آدام این است که ما نمی‌خواهیم آن قدر سریع به هدف برسیم، می‌خواهیم سرعت را کمی برای یک جستجوی دقیق، کاهش دهیم.

$M(t)$ و $V(t)$ مقادیر نخستین گشتاور هستند که به ترتیب میانگین و واریانس گرادیان‌ها است .

فرمول الگوریتم:

$$\theta_{t+1}=\theta_t-\eta/(\sqrt{(v_t+\epsilon)})*M(t).$$

مزیت‌ها:

---

[1] Adadelta

[2] Adam

[3] Momentums

این روش خیلی سریع است و به سرعت همگرا می‌شود.

نرخ یادگیری را یکسوسازی می‌کند، با استفاده از واریانس بالا.

معایب:

از نظر محاسباتی پر هزینه است.

در این پروژه از این بهینه‌ساز استفاده شده‌است زیرا در اکثر پروژه‌ها بهترین عملکرد را نسبت به بقیه‌ی بهینه‌سازها دارد و برای داده‌های پراکنده پیشنهاد شده‌است. سرعت در همگرایی نیز از دیگر نکات در استفاده از این الگوریتم است.

۵-۴- اصطلاحت یادگیری ماشین [۱۷]

ما به این اصطلاحات مثل اپک[1]، اندازه دسته‌ای و تکرار تنها زمانی نیاز داریم که داده‌ها بسیار بزرگ باشند که در تمام پروژه‌ها در یادگیری ماشین جدیدا اتفاق می‌افتد و ما نمی‌توانیم همه داده‌ها را یکجا به کامپیوتر به عنوان ورودی بدهیم. بنابراین، برای غلبه بر این مشکل نیاز است که داده‌ها را به اندازه‌های کوچک‌تر تقسیم کنیم و به کامپیوتر یک به یک دسته‌ها را برای به روز رسانی وزن‌ها شبکه‌های عصبی در پایان هر گام برای متناسب کردن آن با داده‌های داده‌شده، دهیم.

۵-۴-۱- اپک

یک اپک به معنی این است که تمام یک مجموعه فقط یک بار به عنوان ورودی منتقل می‌شود و یک بار خروجی آن دیده شود. از آنجا که یک دور دادن مجموعه داده برای کامپیوتر بسیار بزرگ و پیچیده است, ما آن را در چندین بار به دسته‌های کوچک‌تر تقسیم می‌کنیم.

در شروع ، مشخصا ارائه کل مجموعه داده‌ها به یک شبکه جدید برای یادگیری عملا کافی نیست. باید مجموعه داده کامل را چندین بار به همان شبکه عصبی منتقل کنیم تا از یادگیری شبکه

[1] Epoch

اطمینان حاصل گردد. اما به خاطر داشته باشید که ما از مجموعه داده‌های محدودی استفاده می‌کنیم و برای بهینه‌سازی یادگیری از بهینه‌سازها استفاده می‌کنیم که در ذات خود فرآیند تکراری‌اند. بنابراین به روز رسانی وزن‌ها با یک اپک کافی نیست. الگوریتم دقیقی برای محاسبه تعداد بهینه اپک که برای انواع شبکه‌های عصبی کاربرد داشته باشد، ارائه داده نشده‌است.اما دو پدیده‌ی بیش برازش[1] و کم‌برازش[2] برای تعیین تعداد اپک تعریف شده‌اند.

بیش‌برازش شدن به معنای این است که الگوریتم فقط داده‌هایی که در مجموعه آموزشی[3] یاد گرفته است را می‌تواند به درستی پیش‌بینی کند ولی اگر داده، کمی از مجموعه‌ی آموزشی فاصله داشته باشد، الگوریتمی که بیش‌برازش شده باشد، نمی‌تواند به درستی پاسخی برای این داده جدید پیدا کند.

کم‌برازش شدن نیز زمانی رخ می دهد که الگوریتم یک مدلِ خیلی کلی از مجموعه آموزشی به دست می‌آورد. یعنی حتی اگر خودِ داده‌های مجموعه‌ی آموزشی را نیز به این الگوریتم بدهیم، این الگوریتم خطای قابل توجه خواهد داشت.

تعداد اپک، زمانی که آموزش بین این دو پدیده قرار می‌گیرد که به اصطلاح به آن نقطه‌ی اپتیمال[4] گفته می‌شود، به عنوان بهترین تعداد شناخته می‌شود.

۵-۴-۲- اندازه دسته[5]

همان‌طور که مطرح شد، ارائه کل مجموعه داده‌ها به عنوان ورودی عملا کار مناسب و خردمندانه‌ای به نظر نمی‌رسد. اندازه دسته در حقیقت پارامتری است که با آن تعداد داده‌های دیتاست ک یک جا به شبکه‌ی عصبی داده می‌شود، مشخص شده‌است. توانایی رایانه در محاسبات کامپیوتری و الگوریتم استفاده شده برای بهینه‌سازی شبکه عصبی می‌توانند فاکتورهای مناسبی برای انتخاب

[1] overfitting

[2] underfitting

[3] Train set

[4] Optimal

[5] Batch size

محدوده‌ی اندازه‌ی دسته باشند.در کل ولی پروسه انتخاب کارآمدترین تعداد برای این پارامتر، به صورت آزمون و خطا اتفاق می‌افتد.

۵-۴-۳- تکرار[1]

پارامتر تکرار تعداد دفعاتی را توصیف می‌کند که مجموعه‌ای از داده‌ها از طریق الگوریتم رد می‌شوند. در مورد شبکه‌های عصبی, این به معنای عبور از جلو[2] و عبور برعکس[3] است. بنابراین, هر بار که یک دسته از داده‌ها را از طریق شبکه‌ی عصبی عبور می‌دهید, یک تکرار را به پایان می‌رسانید.

یک مثال ممکن است این پارمتر را روشن‌تر کند. فرض کنید که شما مجموعه داده ۱۰ مثال (یا نمونه) را دارید. شما اندازه دسته ۲ را انتخاب کردید و مشخص کرده‌اید که می‌خواهید الگوریتم به مدت ۳ اپک اجرا شود. بنابراین، در هر اپک شما ۵ دسته دارید (۱۰ / ۲ = ۵). هر دسته از این الگوریتم عبور می‌کند، بنابراین شما ۵ تکرار در هر اپک دارید. از آنجایی که شما ۳ اپک مشخص کرده‌اید، در مجموع ۱۵ تکرار (۵ * ۳ = ۱۵) برای آموزش دارید.

۵-۵- معیارهای ارزیابی شبکه[۴۷]

در این بخش به معرفی معیارهایی جهت ارزیابی شبکه‌ی عصبی می‌پردازیم. از جمله مهم‌ترین عملیات‌ها پس از ساخت و طراحی شبکه ارزیابی عملکرد آن شبکه از نظر دقت و صحت است. برای این منظور معیارهای متفاوتی مطرح شده‌است که هریک به واسطه پارامترهای خود معیارهای متفاوتی برای اندازه‌گیری و مقایسه‌ی کارایی شبکه و الگوریتم‌ها فراهم کردند. ارزیابی، در جهت بهبودبخشی و افزایش کارایی شبکه، امری پر اهمیت است.

[1] Iteration

[2] Forward pass

[3] Backward pass

۵-۵-۱- دسته‌بندی نتایج

این دسته‌بندی نتایج که به صورت جدولی دو در دو باینری است در چهار بخش قابل بررسی است.

۱- True Positive: به معنی مثبت صحیح نشان‌دهنده این است که نتیجه درست شناسایی شده است.

۲- False Positive : به معنی مثبت کاذب نشان‌دهنده این است که نتیجه اشتباه شناسایی شده است.(خطای نوع اول)

۳- True Negative: به معنی منفی صحیح نشان‌دهنده این است که نتیجه به درستی رد شده است.

۴- False Negative : به معنی منفی کاذب نشان‌دهنده این است که نتیجه به اشتباه رد شده است.(خطای نوع دوم)

۵-۵-۲- روش‌های ارزیابی الگوریتم[۴۷]

از تحلیل نتایج به دست‌آمده و جای‌گیری آن‌ها در چهاردسته تعریف شده در بالا، قادر به ارزیابی کیفیت شبکه هستیم. کارایی شبکه نیز برای بهبود عملکرد تفسیرپذیر است. به معرفی شش معیار که معمولا بیشترین استفاده را دارند می‌پردازیم.

۵-۵-۲-۱- ماتریس درهم‌ریختگی[1]

این معیار از ارزیابی به ماتریسی گفته می‌شود که عملکرد الگوریتم را مورد بررسی قرار می‌دهد. در اکثر مواقع این ماتریس برای شبکه‌های یادگیری با نظارت یا با ناظر استفاده می‌گردد، اما در شبکه-های بدون ناظر نیز کاربرد دارد. در این نوع شبکه‌ها به این ماتریس، ماتریس تطابق گفته می‌شود. هر ستون از ماتریس، مقدار پیش‌بینی‌شده را در اختیار قرار می‌دهد و هر سطر از ماتریس وظیفه‌ی نشان

[1] Confusion matrix

دادن واقعیت نتایج را دارد. در خارج از علوم هوش مصنوعی نیز این ماتریس کاربرد دارد ولی به نام‌های ماتریس پیشایندی[1] و یا ماتریس خطا[2] شناخته می‌شود.

۵-۵-۲-۲- دقت[3]

در تعریف کلی، دقت به این معنی است که الگوریتم تا چه حد توانایی پیش‌بینی صحیح خروجی را دارد. با توجه به این معیار می توان به سرعت از درستی آموزش شبکه پی برد. اما این معیار جزئیاتی در باب شبکه در اختیار نمی‌گذارد. نحوه‌ی محاسبه‌ی آن از تقسیم مجموع مثبت صحیح و منفی صحیح بر کل نتایج بدست می‌آید.

فرمول :

$$Accuracy = (TP + TN) / (TP + FN + FP + TN)$$

المان‌های فرمول در بخش ۵-۵-۱ تعریف شده‌اند.

۵-۵-۲-۳- صحت[4]

زمانی که پارامتر مثبت کاذب بالا باشد، معیار صحت کاربردی تلقی می‌شود. فرض کنید برای شبکه‌ای که وظیفه‌ی تشخیص سرطان را دارد، صحت پایین است. تعداد افرادی که به اشتباه بیمار تلقی می‌شوند، زیاداند. این نتایج استرس و هزینه‌ی زیادی را برای فرد به اشتباه دارای سرطان رقم می‌زند. صحت در حقیقت مقدار، مواردی که درست در کلاس طبقه‌بندی شده‌اند را نسبت به کل نتایجی که چه به غلط و چه به درست در همان کلاس قرار گرفته را مورد بررسی قرار می دهد. زمانی از این پارامتر استفاده می‌شود که هدف دستیابی به دقت در تشخیص نمونه صحیح در کلاس صحیح است.

فرمول:

---

[1] Contingency matrix

[2] Error matrix

[3] Accuracy

[4] Precision

Precision = TP / (TP + FP)

المان‌های فرمول در بخش ۵-۵-۱ تعریف شده‌اند.

۵-۵-۲-۴- فراخوانی[1]

در مواقعی ممکن است دقت در تشخیص کلاس غلط یا منفی اهمیت داشته‌است. در مواقعی که پارامتر منفی کاذب مقدار قابل ملاحظه‌ای را دارد، معیار فراخوانی لازم به بررسی است. اگر همان شبکه‌ی تشخیص سرطان را مورد مطالعه قرار دهیم درمی‌یابیم که اگر منفی کاذب بالا باشد یعنی این الگوریتم افراد زیادی که آلوده به سرطان هستند را به غلط سالم دسته‌بندی می‌کند. فراخوانی نسبت تعداد صحیح تشخیص داده شده به تعداد نتایج واقعی دارای آن ویژگی است.

فرمول:

Recal = Sensitivity = TP / (TP + FN)

المان‌های فرمول در بخش ۵-۵-۱ تعریف شده‌اند.

۵-۵-۲-۵- F-measure/F1 Score

این معیار در حقیقت سنجشی از دقت است. وابسته به فراخوانی و صحت است. در حقیقت F1 score میانگین هارمونیک میان دو پارامتر مطرح شده‌است. که در بهترین شرایط معادل عدد یک و در بدترین شرایط معدل عدد صفر است.

فرمول:

F-measure = 2*(Recall*Precision) / (Recall + Precision)

[1] Recall

۵-۵-۲-۶- خاصیت[1]

خاصیت از متداول‌ترین پارامترها که معمولا در کنار پارامتر حساسیت بررسی می‌شود، است. در معنی کلی نسبت نتایج که به درستی رد شده‌اند یا به روایتی دیگر نمونه منفی درست تشخیص داده شده به کل نمونه های منفی واقعی چه درست و چه غلط تشخیص داده شده، است.

فرمول :

$$Specificity = TN /(TN + FP)$$

المان‌های فرمول در بخش ۵-۵-۱ تعریف شده‌اند.

۵-۵-۲-۷- MCC[2]

معیار دیگری برای عملکرد شبکه‌های یادگیری ماشین استفاده می‌شود، MCC است. این پارامتر مقادیری بین منفی یک تا یک را می‌گیرد. که مثبت یک نشان‌دهنده‌ی پیش‌بینی بدون خطا و با دقت الگوریتم است. منفی یک نشان‌دهنده‌ی این است که موارد پیش‌بینی شده و واقعیت نتایج باهم یکسان نیستند. عدد صفر این احتمال را می دهد که نتایج به صورت کاملا تصادفی پیش‌بینی شده‌اند.

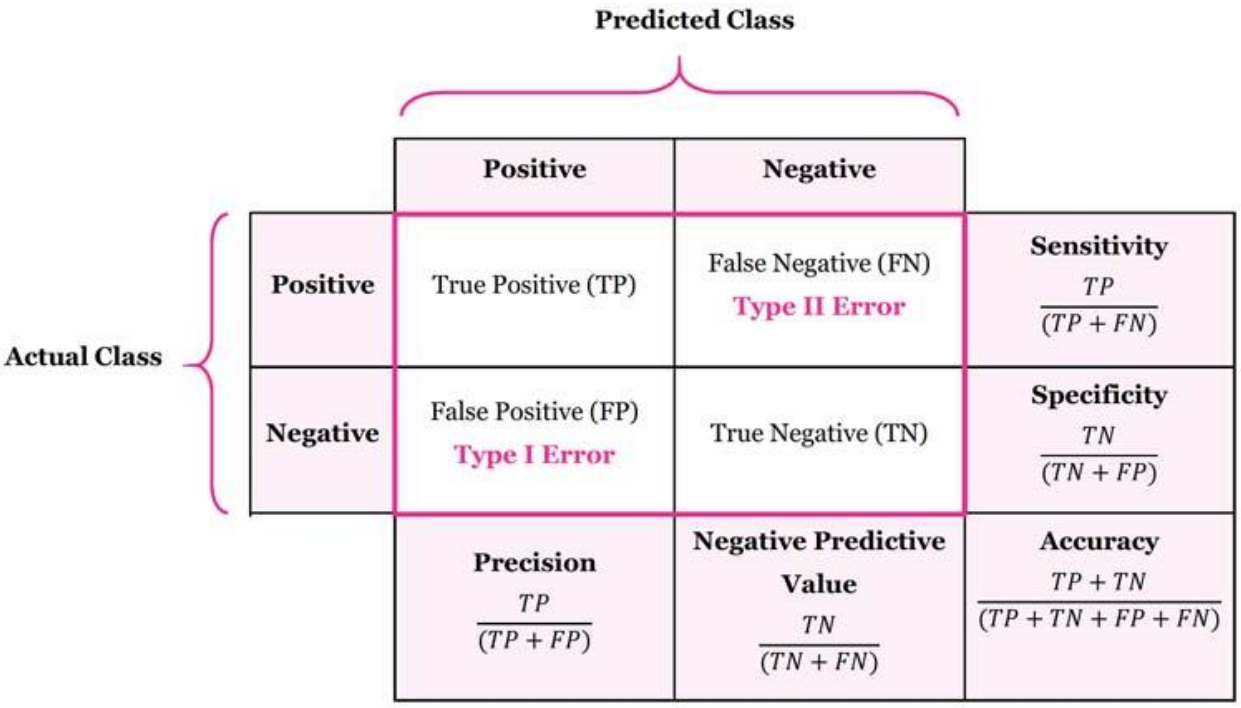
فرمول:

$$MCC = (TP*TN-FP*FN)/ \sqrt{((TP + FP)(TP + FN)(TN + FP)(TN + FN))}$$

المان‌های فرمول در بخش ۵-۵-۱ تعریف شده‌اند.

[1] Specificity

[2] Matthews Correlation Coefficient



شکل ۵-۲- معیارهای ارزیابی شبکه‌ی عصبی[۴۷]

در شکل ۵-۲ نحوه‌ی محاسبه‌ی معیار های تعریف شده به صورت یک جدول کامل نمایش داده شده‌است. ۵ معیار در کادرهای صورتی رنگ در پایین و سمت راست جدول قرار دارند.

۵-۵-۳- نرخ nام خطا[۱][۴۷]

نرخ خطای اول (Top-1 error) اصطلاحی است که برای توصیف دقت یک الگوریتم برای طبقه‌بندی استفاده می‌شود. که نشان دهنده‌ی احتمال صحیح بودن اولین حدس پیش‌بینی شبکه است.

معمولاً طبقه‌بندی کننده، احتمال یا مقدار اطمینان برای هر کلاس را ارایه می‌دهد. مثلا: یک گذاره به شکل "من ۹۰ درصد مطمئن هستم که این تصویر یک حیوان است"،و گذاره‌های دیگر "من ۰/۱ درصد مطمئن هستم که این تصویر یک انسان است"، و غیره).

[1] Top-N Error

اگر حدس بالا صحیح باشد پاسخ صحیح برای نرخ اول در نظر گرفته می‌شود (به عنوان مثال، بالاترین احتمال برای طبقه "حیوان" است و تصویر تست در واقع یک حیوان است).

اگر حداقل یکی از n حدس دسته‌بندی کننده، درست باشد، پاسخ صحیح در سطح nام در نظر گرفته می‌شود.(Top-N Error)

کتابخانه‌ی scikit learn روش accuracy_score را برای ارزیابی دقت مدل معرفی کرده است. پارامتر score دقت طبقه‌بندی را مشخص می‌کند.

۵-۵-۴- منحنی ROC[1]

یک منحنی مشخصه عملکرد که به اختصار آن را منحنی ROC می‌نامیم، یک نمودار برای نمایش توانایی ارزیابی یک سیستم دسته‌بندی باینری محسوب می‌شود که آستانه تشخیص آن نیز متغیر است. منحنی ROC، نرخ مثبت صحیح یا (True Positive Rate) که به اختصار TPR نامیده می‌شود را برحسب نرخ مثبت کاذب یا (False Positive Rate) با نام اختصاری FPR، ترسیم می‌کند. البته توجه داشته باشید که آستانه برای این مقادیر، متغیر است. به همین دلیل، یک نمودار پیوسته ایجاد خواهد شد.

بنابراین فضای ROC بوسیله این دو شاخص یعنی FPR روی محور افقی و TPR روی محور عمودی شکل داده می‌شود. توجه داشته باشید که هر عنصر از ماتریس درهم‌ریختگی یک نقطه در منحنی ROC را تشکیل می‌دهد.

با استفاده از دستور roc_curve که کتابخانه‌ی scikit learn ایجاد کرده است می‌توان از امکانات این منحنی استفاده کرد.

[1] Receiver operating characteristic curve

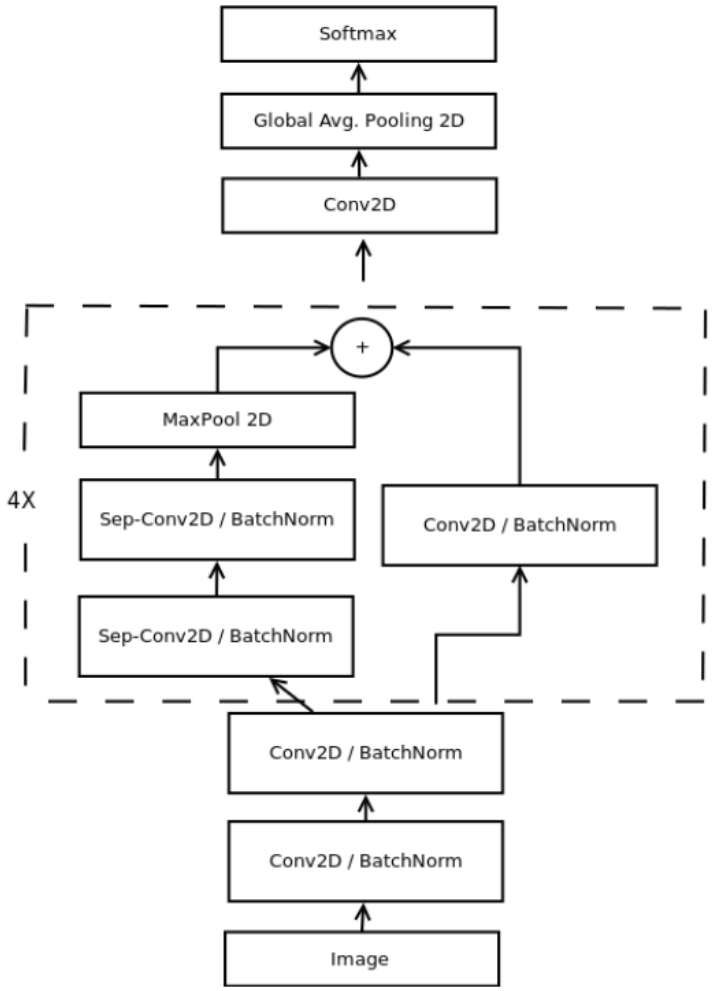
AUC[1] -۵-۵-۵

پارامتر دیگر مقدار زیر سطح نمودار ROC است. مقدار عددی AUC به وضوح عددی بین صفر تا یک است و نشان می‌دهد قدرت تشخیص یا درستی نتایج یک آزمون چقدر می‌باشد. اگر این عدد به یک نزدیک باشد، به معنای آن است که داده‌ها عموماً در بالای خط نیم‌ساز قرار گرفته‌اند و میزان نرخ مثبت صحیح بالا است و مدل از قدرت تشخیص مناسبی برخوردار است. اعداد AUC نزدیک به ۰/۵ همان برابری نرخ مثبت صحیح و نرخ مثبت کاذب را نشان می‌دهد و اعداد کمتر از ۰/۵ بیانگر بالاتر بودن نرخ مثبت کاذب در مقایسه با نرخ مثبت صحیح است.

۵-۶- پیاده‌سازی

هدف از پروژه بهینه‌سازی معماری‌های ساخته شده برای تشخیص احساسات است. در این زمینه تعداد قابل توجه‌ای از معماری‌ها کاربرد داردن از جمله رس‌نت و VGG اما به علت پیچیدگی و فراهم کردن حجم زیادی از پارامترها و ویژگی‌ها، برای این پروژه پیشنهاد نمی‌شوند. زیرا سرعت در تشخیص احساسات یکی از معیارهای حائز اهمیت است. با بررسی انواع معماری‌ها از نظر سرعت پاسخگویی و دقت، معماری الهام گرفته شده از معماری اکسپشن انتخاب شده‌است. تا حدودی از آن ساده‌تر است و در تعداد پارامتر نیز کمتر است. در این راستا به طراحی این معماری و یادگیری آن با دیتاست FER پرداختیم در ادامه نتایج دقت و ماتریس به‌هم‌ریختگی آن نمایش داده شده‌است. این یادگیری با بهینه ساز آدام انجام شده‌است. در بستر کلب شرکت گوگل به آموزش گذاشته شده‌است هر اپک حدود ۲۴۱ ثانیه زمان برده‌است.(باتوجه به سرعت ۲۶۸ میلی ثانیه برای هر مرحله) ۱۰۰ اپک برای یادگیری در نظر گرفته شده‌است. اندازه دسته‌های ۳۲تایی برای هم سرعت و دقت یادگیری مناسب در نظر گرفته شده‌است.

[1] Area Under the ROC curve

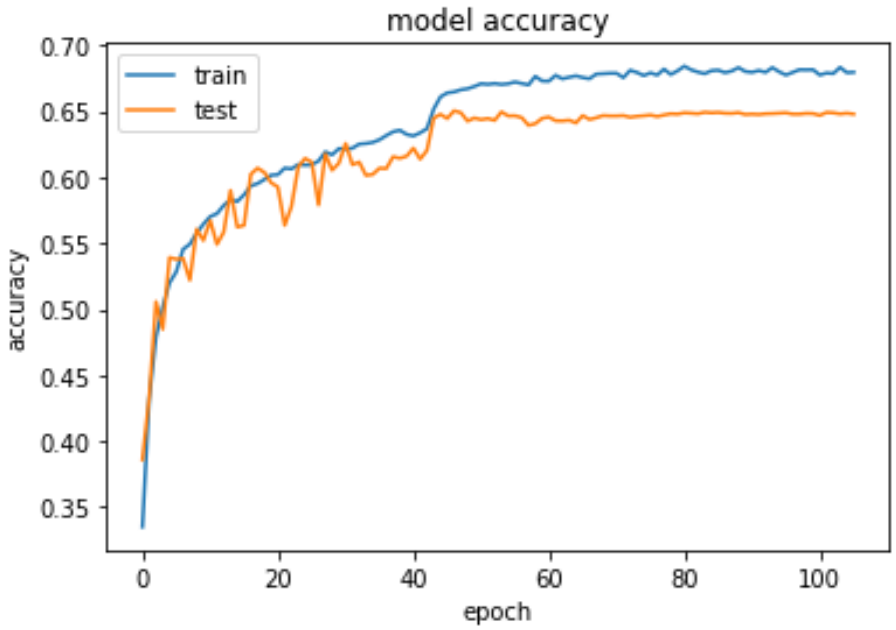


شکل ۵-۳- مدل الهام گرفته‌شده از اکسپشن

همانطور که در شکل ۵-۳ معماری استفاده شده برای پژوهش اولیه‌ی این پروژه مطرح شده است، مشخص است که در این تحقیق از لایه‌ی پولینگ و میانگین‌گیر جهانی با تابع فعال‌ساز سافت-مکس که خروجی آن به اندازه‌ی تعداد کلاس‌های موجود در دیتاست(۷ کلاس) است، استفاده شده است. بعد از پایان فرآیند آموزش، مدل به منظور استفاده‌های بعدی ذخیره می‌شود. اگر خلاصه‌ای از مدل ساخته شده فراهم کنیم در انتها به ۵۸۴۲۳ پارامتر دست یافته‌است که از این تعداد ۱۴۷۲ پارامتر قابل آموزش نیستند. در ادامه مدل با دیتاست برای محاسبه دقت و اعتبارسنجی زیان[1] ارزیابی شده‌است.

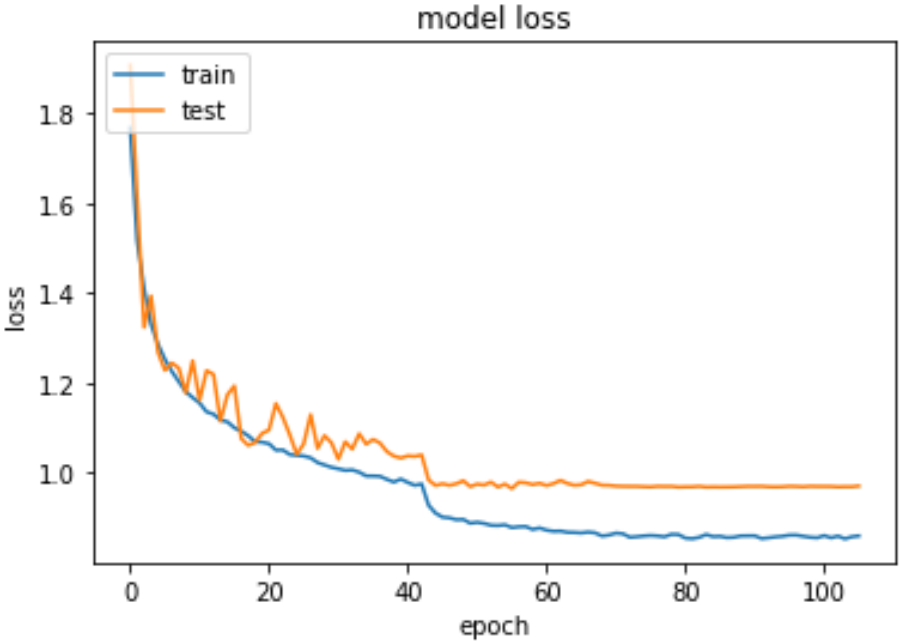
[1] Validation Loss

هدف این است که برچسب‌های داده‌های مجموعه‌ی تست، به کمک مدل آموزش دیده بر روی داده های مجموعه‌ی آموزش، پیش‌بینی شود تا در نهایت با مقایسه‌ی آن‌ها با برچسب‌های اصلی این مجموعه، بتوان عملکرد مدل را به کمک پارامترهای دقت، اعتبارسنجی زیان و ماتریس درهم ریختگی ارزیابی کرد.



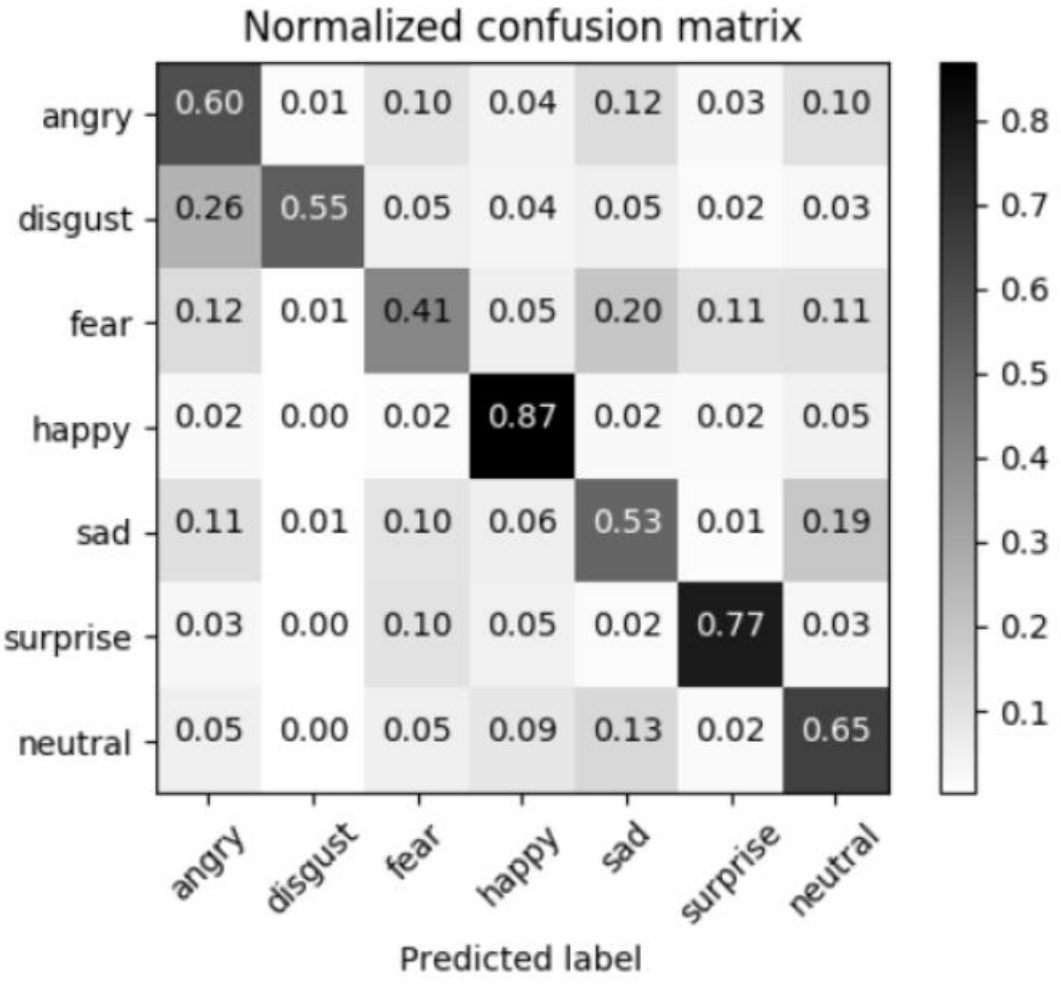
شکل ۵-۴- نمودار دقت برای مدل اصلی

همانطور که در شکل ۵-۴ قابل مشاهده‌است، مدل در آموزش به دقتی در حدود ۰/۶۷۵ رسیده‌است و در ارزیابی (test) به دقتی در حدود ۰/۶۴۰ رسیده است.



شکل ۵-۵- نمودار اعتبارسنجی زیان برای مدل اصلی

در شکل ۵-۵- قادر به مشاهده اطلاعاتی از جمله‌ی مقادیر تابع زیان در انتها برای داده‌های آموزش به عدد ۰/۹۸۹ و برای داده‌های تست به عدد ۰/۸۷۳ منتهی شده‌است. این آموزش حدود هفت ساعت و دوازده دقیقه زمان برد در حقیقت هر مرحله اپک به طور میانگین ۲۸۰ ثانیه زمان برده‌است.



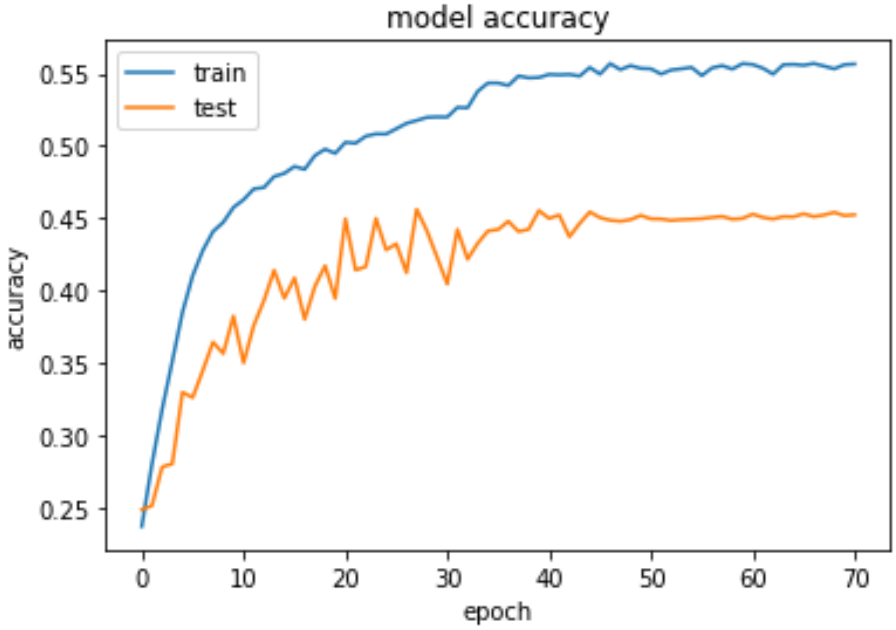
شکل ۵-۶- ماتریس درهم‌ریختگی نرمال شده

در شکل ۵-۶ ماتریس درهم‌رختگی نرمالیز شده یعنی بین ۰ تا ۱ شده نمایش داده شـده‌اسـت. خوشحالی و تعجب از حالت‌های با بالاترین مقادیر صحت در تشخیص هستند، در حالی که ترس با ۰/۴۱ کمترین مقدار را دارد.

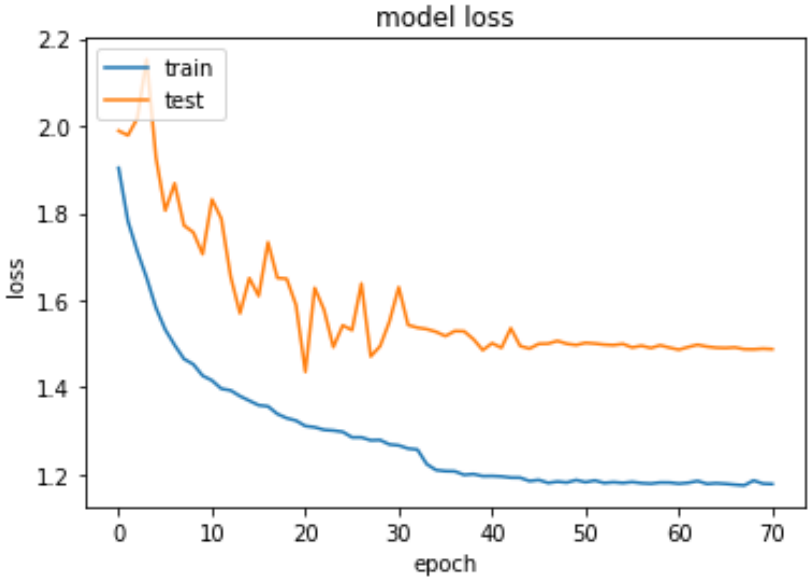
۵-۶-۱- افزایش پارامترها

در این مرحله جهت آموزش بهتر و افزایش کارایی و دقت مدل در این معماری سعی بر افزایش پارامترهای مدل شد. در این باب در جهت بهینه بودن و ارتقای معماری بخشی از آن که در حقیقت وظیفه‌ی پارامترسازی و جداسازی ویژگی‌ها بود را پیچیده‌تر کرده‌ایم.

افزایش یک سیکل به ۴ بار تکرار در معماری، به پیچیدگی و همچنین تعداد پارامترهای این مدل افزود. تغییرات حائز اهمیتی از جمله افزایش تعداد پارامترها به ۲۰۴۰۸۷ افزایش یافت که تقریبا حدودا ۳/۵ برابر شده‌است نسبت به مدل مطرح شده‌است. البته در این خصوص حدود سه هزارتا از این پارامترها غیرقابل آموزش است.



شکل ۵-۷- نمودار دقت برای مدل افزایش پارامتر داده‌شده



شکل ۵-۸- نمودار اعتبارسنجی تابع زیان برای مدل افزایش پارامتر داده‌شده

همان‌طور که از تحلیل نمودار های دقت در شکل ۵-۷ و تابع زیان در شکل ۵-۸ مشخص است با اینکه به تعداد پارامترهای مدل جهت آموزش دقیق‌تر این معماری افزوده‌ایم، دقت این معماری به حدود ۰/۵۶۳۴ در آموزش رسیده است و در تست به دقت پایینی در حدود ۰/۴۳۷ رسیده‌است. با مقایسه مقادیر به دست آمده با مدل اصلی می توان دریافت که این افزایش پارامتر سبب پیچدگی بیش از حد مدل شده‌است و نتیجه قابل قبولی ارائه نشده‌است.

۵-۶-۲- بهبودسازی مدل پیچیده

زمانی که مدل شما روی داده آموزش خیلی خوب عمل کند ولی روی داده تست خوب عمل نکند، در این حالت مدل بیش برازش شده و بیش از حد روی تک تک داده های آموزش برازش شده است. در تحقیقات انجام شده برای افزایش دقت در مدل‌های پیچیده که معمولا دچار کم‌برازش یا بیش-برازش است روش های زیادی ارائه شده‌است.

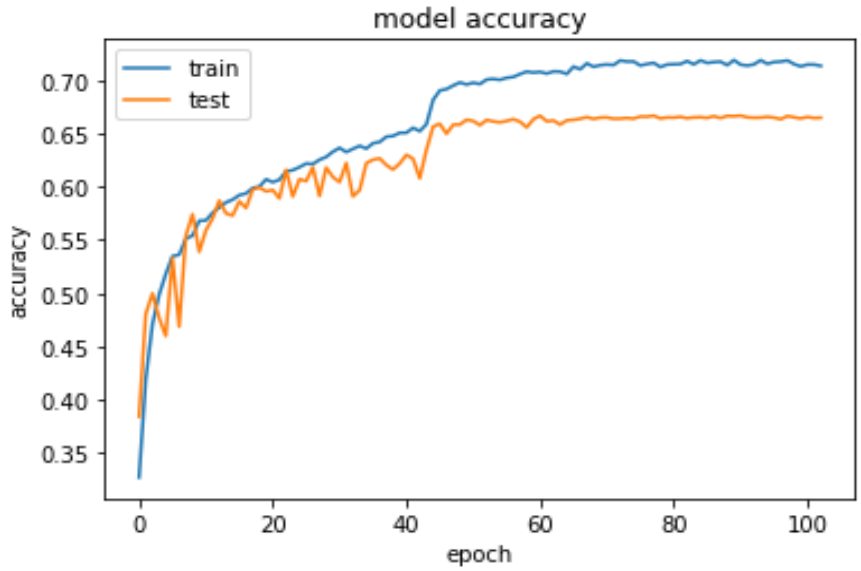
رگولاریزاسیون [1] با کمک تکنیک‌های مختلف، مدل را مجبور می‌کند که از پیچیدگی دوری کرده و تا جایی که میتواند ساده‌تر باشد. رگولاریزاسیون به نوع مدلی که استفاده می‌کنیم بستگی دارد. دراپ‌اوت[2] یک رویکرد برای رگولاریزاسیون در شبکه‌های عصبی است که باعث کاهش یادگیری‌های تکراری میان نورون‌ها می‌شود. دراپ‌اوت به معنای کنار گذاشتن بخش‌هایی (units) از یک شبکه عصبی است. یک شبکه عصبی که شامل تعدادی نورون است، در نظر بگیرید. دراپ‌اوت به این معنا است که در حین آموزشِ این نورون‌ها، از تعدادی از آن‌ها به صورت تصادفی چشم‌پوشی می‌شود. چشم‌پوشی یعنی اینکه آن نورون‌های خاص، در مسیر رفت یا برگشت در نظر گرفته نمی‌شوند.

اگر بخواهیم فنی‌تر بررسی کنیم، دارپ‌اوت یعنی اینکه در هر مرحله از آموزش، نودهایی از شبکه، با احتمال p-۱ کنار گذاشته شده و نودهای دیگری با احتمال p، حفظ می‌شوند. بنابراین یک شبکه کاهش یافته باقی می‌ماند

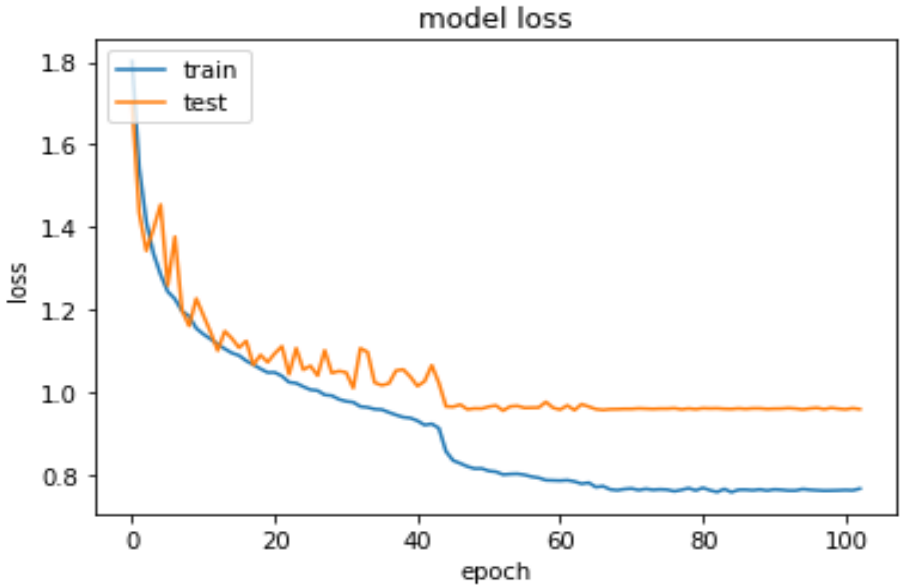
[1] Regularization

[2] DropOut

در نتیجه برای بهینه‌سازی این مدل از دراپ‌اوت استفاده شده‌است و همچنین برای کاهش پیچدگی مدل از تابع دنس[1] در کتابخانه کرنل در جهت کاهش نورون های پدید آمده در مرحله‌ی پنجم تکرار که برای افزایش تعداد پارامترها در این پروژه اضافه شده‌است، استفاده کرده‌ایم.



شکل ۵-۹- نمودار دقت برای بهبودیافته‌ی مدل افزایش پارامتر داده‌شده



شکل ۵-۱۰- نمودار اعتبارسنجی تابع زیان برای بهبودیافته‌ی مدل افزایش پارامتر داده‌شده

[1] Dense

در شکل‌های ۵-۹ و ۵-۱۰ پی به ارتقا در دقت و کاهش تابع زیان نسبت به مدل اصلی می‌بریم که از هدف‌های دست‌یافته در طی تغییرات انجام شده در مدل است. افزایش پارامتر های قابل آموزش در کنار دقت بالاتر به کارایی بهتر مدل منجر شده‌است. دقت نهایی در این معماری به ۰/۷۱۳ در آموزش و ۰/۶۶۴ در تست برای معیار دقت رسیده‌ایم و همچنین با کاهش تابع زیان به عدد ۰/۷۶۶ از آموزش بهتر شبکه آگاه شده‌ایم.

فصل ششم

جمع‌بندی و نتیجه‌گیری و پیشنهادات

جمع‌بندی و نتیجه‌گیری

توسعه در حوزه‌ی تکنیک‌های یادگیری عمیق و شبکه‌های عصبی در هر زمینه از جمله شناسایی احساسات و چهره برای طبقه‌بندی افراد با پیشرفت‌های چشم‌گیری همراه بوده‌است. این پیشرفت‌ها در خصوص بازاریابی و شناسایی بهتر و درک افراد و احساسات آن‌ها منجر به دقت و سرعت لحظه‌ای بالایی شده‌است که در بعضی زمینه‌ها حتی از انسان پیشی گرفته‌است. در حوزه ربات و ارتباط انسان با آن این مبحث نیاز به پوشش در جبهه‌های مختلف از جمله سن، جنسیت و قومیت دارد. البته لازم به ذکر است که استفاده از مدل‌هایی مناسب با توجه به مشخصات داده‌های موجود در دیتاست برای دستیابی به نتایج قابل قبول بسیار مهم است. اقداماتی مبنی بر آماده‌سازی مجموعه داده میتواند عملکرد شبکه‌ها را بهبود ببخشد. در این پروژه با مدل ارائه شده که الهام گرفته از معماری اکسپشن است، هدف کلی که در مرحله‌ی اول تشخیص چهره و در مرحله بعدی تشخیص احساسات در ۷ دسته که با معیارهای کلی فلسفه‌ی اخلاق مشخص شده‌است، به دست آمده‌است. همچنین استفاده از دیتاست FER نیز در به حقیقت پیوستن این خواسته با فراهم کردن تنوع بالا و برچسب‌گذاری درست کمک کرد. این مدل به دقتی در حدود ۶۷٪ ، در چالش ایمجنت توانست مقام کسب کند و ثبت جهانی شد. در این پروژه با افزایش پارامتر، موفق به دست‌یابی به دقتی در حدود ۷۱٪ شد. که دقت بهتر و کارایی بالاتری برای این معماری رقم زد.

بنابراین میتوان نتیجه گرفت که بهره‌گیری از مدل‌های پیچیده شبکه‌های عصبی در تشخیص احساسات در کنار بهینه سازی آن بسیار مفید است. این مقوله هم در زمینه‌ی ارتباط انسان‌ها با ربات‌ها و هم در زمینه بازاریابی بسیار اهمیت دارد.

پیشنهادات

برای آموزش بهتر در یادگیری ماشین نمی‌توان از تاثیر دیتاست چشم‌پوشی کرد همانطور که در مقاله ذکر شده‌است دیتاست‌های فراوانی در باب تشخیص احساسات از چهره در دسترس است. برچسب-گذاری بدون صحت و نامتقارن بودن و حتی عدم تطبیق اندازه و کیفت تصاویر از جمله مشکلات این دیتاست‌ها اند. در آینده با فراهم کردن مجموعه‌ی داده‌های کامل‌تر می‌توان به کاربردی کردن مدل آموزش دیده کمک کرد و به دقت آن افزود.

در حوزه‌ی یادگیری نیز استفاده از منابع سخت‌افزاری قوی‌تر برای سرعت‌دهی به فرآیند از پیشنهادات قابل توجه است. در راستای تکرار مکرر عملیات یادگیری برای ارتقای دقت شبکه، این یادگیری با سیستم‌هایی دارای سخت‌افزار ضعیف بسیار زمان‌بر است.

در تلاش برای کامل کردن این شبکه که در نهایت برای شناسایی بهتر انسان‌ها طراحی شده است می‌توان جنبه‌های مختلفی را در نظر گرفت. به عنوان مثال افزایش برچسب‌ها در خصوص ظریف‌تر کردن طبقه‌بندی‌های شبکه و یا اهمیت بر تشخیص هم زمان جنسیت و سن در کنار احساسات به کامل‌تر شدن هدف این پروژه کمک خواهد کرد. نمی توان از تاثیر جنسیت و سن در تشخیص احساسات صرف نظر کرد پس با حساسیت در این المان‌ها می‌توان به دقت بالاتری برای این پروژه دست یافت.

منابع و مراجع

1. D. Keltner and P. Ekman, "Facial Expression of Emotion. In M. Lewis, & J. Haviland-Jones (Eds.)", Handbook of Emotions, New York: Guilford Publications Inc, 2000.
2. Russell and J. Fenmandez-Dols, "The psychology of facial expression", Cambridge University Press, 1997.
3. J. Beckmann and D. Lew, "Reconciling evidence-based medicine and precision medicine in the era of big data: challenges and opportunities", Genome Med, 2016.
4. G. Weber, K. Mandl and I. Kohane, "Finding the missing link for big biomedical data", Jama, 2014.
5. C. Loconsole, D. Chiaradia, V. Bevilacqua and A. Frisoli, "Real-time emotion recognition: an improved hybrid approach for classification performance", Intelligent Computing Theory, 2014, pp. 31-320.
6. J. Hirschberg and C.D. Manning, "Advances in natural language processing", Oxford University Press, in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, System Demonstrations, Stroudsburg, PA, 2015, pp. 261-265.
7. Y. LeCun, Y. Bengio and G. Hinton, "Deep learning. Nature", Department of Computer Science, University of Toronto, Ontario, Canada, 2015, pp. 436-444.
8. O. Russakovsky et al. "International Journal of Computer Vision". ImageNet Large Scale Visual Recognition Challenge, vol.43, 2015, pp. 211-252.
9. I. Goodfellow et al. "Challenges in Representation Learning: A report on three machine learning contests", Université de Montréal, Montréal, Canada 1 Jul 2013
10. G. Zhao and M. Pietikainen, "Dynamic Texture Recognition Using Volume Local Binary Patterns", Machine Vision Group, Infotech Oulu and Department of Electrical and Information Engineering, Dynamical Vision, 2005.
11. P. Das, H. Behera, S. Pradhan, H. Tripathy and P. Jena, "A modified real time algorithm and its performance analysis for improved path planning of mobile robot,

In Computational intelligence in data mining, Springer India, vol. 2, 2015, pp. 34-221.

12. Nilsson andJ. Nils, "The Quest for Artificial Intelligence", Cambridge University Press, Cambridge, 2009.
13. M. De Leeuw and K. Bergstra, "The History of Information Security: A Comprehensive Handbook" 2007, pp. 266.
14. Gates and A. Kelly, "Our Biometric Future: Facial Recognition Technology and the Culture of Surveillance", NYU Press, Newyork, the US, 2011, pp.48–50
15. K. Kundu, S. Mitra, D. Mazumdar and K. Pal, "Perception and Machine Intelligence", In First Indo-Japan Conference, PerMIn 2012, Kolkata, India, 12-13 Jan 2011.
16. K. Varagu, "Cars May Soon Warn Drivers Before They Nod Off", Huffington Post, 2017.
17. M. Mohri, A. Rostamizadeh and A. Talwalkar, "Foundations of Machine Learning", the MIT Press, 2012.
18. M. Van Otterlo and M. Wiering, "Reinforcement learning and markov decision processes", Katholieke Universiteit Leuven, Belgium, 2012, pp. 3–32.
19. M. Valueva,N. Nagornov, P. Lyakhov, G. Valuev and N. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation", Mathematics and Computers in Simulation, May 2020.
20. K. Fukushima, "Neocognitron", NHK Science and Technical Research Laboratories Japan, Scholarpedia, Japan, 2007.
21. D. Hubel, H. Wiesel, "Receptive fields of single neurones in the cat's striate cortex", J. Physiol, 1954, pp: 148-574.
22. A. Azulay, Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?", in Journal of Machine Learning Research, TheUk, 2019.
23. M. Singh and D. Sahu, "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation", In International journal for research, vol.5, LISA Lab, 31 August 2013.

24. H. Habibi Aghdam and E. Jahani Heravi, "Guide to convolutional neural networks: a practical application to traffic-sign detection and classification", Springer International Publishing, Switzerland, 2017.
25. D. Scherer, A. Müller, C. Andreas and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition",in 20th International Conference on Artificial Neural Networks (ICANN), Thessaloniki, Greece, 2010, pp.92–101.
26. J. Han, M. Jun and Claudio."The influence of the sigmoid function parameters on the speed of backpropagation learning", Springer Berlin Heidelberg, 1995, pp.195-201.
27. Y. LeCun, L. Bottou, B. Orr and K. Müller, "Efficient BackProp", In G. Orr Neural Networks: Tricks of the Trade, Springer, 1998.
28. A. L. Maas, Y. Hannun and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models", Computer Science Department, Stanford University, CA, 2014.
29. D. Ciresan, U. Meier, J. Masci, M. Gambardella and J. Schmidhuber, "Flexible, High Performance Convolutional Neural Networks for Image Classification", 17 November 2013.
30. S.S. Farfade, M. Saberian and L. Li, "Multi-view Face Detection Using Deep Convolutional Neural Networks", Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, April 2015.
31. P. Burkert ET AL."DEXPRESSION: Deep Convolutioal Neural Network For Expression Recognition", German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany, 2017.
32. J. Chen ET AL."Recognizing Emotion from static images", Stanford university department of computer science, 2016.
33. Viola and Jones, "Robust Real-time Object Detection", IJCV 2001, pp: 1-31.
34. F. Fleuret and D. Geman, "Coarse-to-fine face detection", in Int J. Computer Vision, January 2001, pp:81-107.

35. R. Rojas, "AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting", Freie University, Berlin, 2009.
36. A. Krizhevsky, Sutskever, H. Ilya and E. Geoffrey, "ImageNet classification with deep convolutional neural networks", Communications of the ACM, 2017, pp: 84–90.
37. K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", University of Oxford, UK, 2014.
38. C. Szegedy, W. Liu, Y. Jia, P. Sermanet and S. Reed, "Going Deeper with Convolutions", University of Michigan and University of North Carolina, 2015.
39. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture", Computer Vision University College London, 2016.
40. R. Srivastava, K. Greff and J. Schmidhuber, "Training Very Deep Networks", 2015.
41. Franc and O. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions", Google Inc, 2015.
42. C. Szegedy, S. Ioffe, V. Vanhoucke and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning", Google Inc, 2016.
43. https://paperswithcode.com/sota/image-classification-on-imagenet , Available on: July 2021
44. https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d, Available on: August 2021
45. https://b.fdrs.ir/15i , Available on: August 2021
46. L. Bottou, F. Curtis and J. Nocedal, "Optimization Methods for Large-Scale Machine Learning", Society for Industrial and Applied Mathematics, 2016.
47. https://bigdata-ir.com, Available on: July 2021

پیوست‌ها

در این بخش قسمتهای مختلف کد، مرحله به مرحله توضیح داده شده است.

ابتدا فایا دیتاست به نام 'fer2013.csv' را در پوشه ای به نام 'emotiondetection' در درایو گوگل آپلود شده‌است.

سپس با کدهای رو به رو به آن دسترسی پیدا کردیم:

```
-from google.colab import drive
-drive.mount('/content/drive')
-cd /content/drive/My Drive/emotiondetection
-dataset_path = 'fer2013.csv'
```

داده های داخل مجموعه داده ۴۸ پیکسل در ۴۸ پیکسل است:

```
-image_size = (۴۸،۴۸)
```

حال برای بارگذاری داده های مجموعه داده در آرایه های مشخص، تابع loadfes را تعریف کردیم.

```
-import pandas as pd
-import cv2
-import numpy as np
def loadfer():
    data = pd.read_csv(dataset_path
    pixels = data['pixels'].tolist()
    width, height = 48, 48
    face = []
    for pixel_sequence in pixels:
        face = [int(pixel) for pixel in pixel_sequence.split(' ')]
        face = np.asarray(face).reshape(width, height)
```

```
        face = cv2.resize(face.astype('uint8'),image_size)
        faces.append(face.astype('float32'))
    faces = np.asarray(faces)
    faces = np.expand_dims(faces, -1)
    emotions = pd.get_dummies(data['emotion']).values
```

در انتهای این تابع، داده های تصویری چهره ها و داده های حالات احساسی دسته بندی می شوند. حال برای بهینه سازی داده‌های چهره، آرایه ها را نرمال و gray scale می کنیم با اسفاده از تابع preprocess_input که در ادامه تعریف شده است.

```
def preprocess_input(x, v2=True):
    x = x.astype ('float32')
    x = x / 255.0
    if v2:
        x = x - 0.5
       x = x * 2.0
    return x
```

حال با صدا کردن دو تابع تعریف شده از آن ها بهره می بریم:

```
faces, emotions = loadfer()
faces = preprocess_input(faces)
```

داده های تست و آموزش را برای یادگیری شبکه عصبی و بی صورت تصادفی (shuffle) به نسبت ۰/۲ انتخاب می کنیم:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(faces, emotions,test_size=0.2,shuffle=True)
```

پارامتر ها:

```
batch_size = 32
num_epochs = 100
input_shape = (۱ ,۴۸ ,۴۸)
num_classes = 7
```

batch size: برای جلوگیری از سپری کردن زمان طولانی و اختصاص دادن تک به تک داده ها به شبکه عصبی داده ها را به صورت بچ (batch) های گروهی از تصاویر ۳۲تایی به شبکه عصبی خواهیم داد.

num_epochs: تعداد دفعات یادگیری هست. هرچه بالاتر برود بهتر است تا زمانی که overtrain نشود.

Input shape: نشان دهنده تعداد پیکسل ها در طول و عرض تصاویر است که برابر ۴۸ پیکسل است و عدد ۱ نشان دهنده تک رنگ بودن تصاویر است (gray scale).

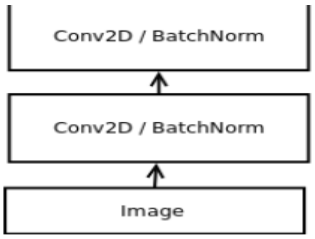
Num class: نشان دهنده تعداد کلاس های تفکیک شده برای حات های چهره است که در پیش گزارش قبل تعریف شده‌اند.

کدهای مرتبط به معماری :

```
img_input = Input (input_shape)
regularization = l2(۰,۰۱)
```

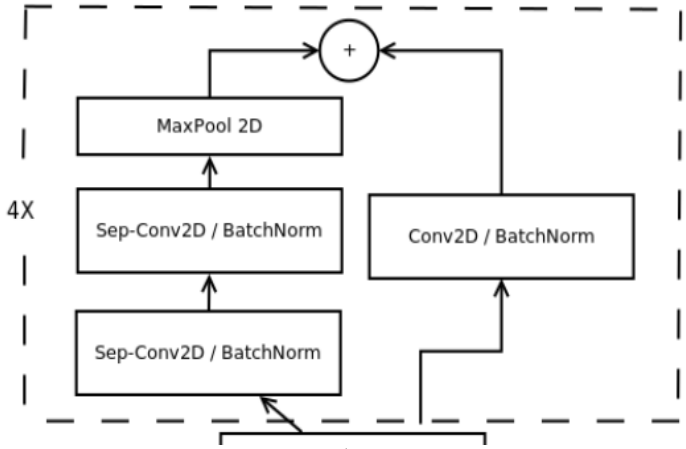
داده های تصویری را وارد می‌کنیم و کد regularization برای overfit هست که در ادامه توضیح داده خواهد شد.

```
x = Conv2D (8, (3, 3), strides= (1, 1), kernel_regularizer=regularization,
use_bias=False) (img_input)
x = BatchNormalization () (x)
x = Activation ('relu') (x)

x = Conv2D (8, (3, 3), strides= (1, 1), kernel_regularizer=regularization,
use_bias=False) (x)
x = BatchNormalization () (x)
x = Activation ('relu') (x)
```



شکل ۱ :قسمت اول معماری شبکه

کانولوشن دارای ۸ نرون است با فیلتر ۳در۳ و با حد فاصله ی تک پیکسل اجرا می‌شود. کد های بالا قسمتی از معماری شبکه را به عهده می‌گیرند که در شکل ۱ در بالا مشاهده می‌کنید.



شکل ۲

برای قسمتی از معماری شبکه که در شکل۲ مشاهده می کنید ابتدا قسمت سمت راست را به صورت زیر تعریف می‌کنیم:

```
residual = Conv2D(16, (1, 1), strides=(2, 2), padding='same', use_bias=False)(x)
residual = BatchNormalization()(residual)
```

حال طرف چپ را می‌سازیم و از separable conv استفاده کردیم تا پارامتر ها را کاهش دهیم:

```
x = SeparableConv2D(16, (3, 3), padding='same',
kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
padding same size stride 1,1
x = SeparableConv2D(16, (3, 3), padding='same',
kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)
x = MaxPooling2D ((3, 3), strides= (2, 2), padding='same') (x)
```

در این مرحله باید دو داده را با هم جمع کنیم:

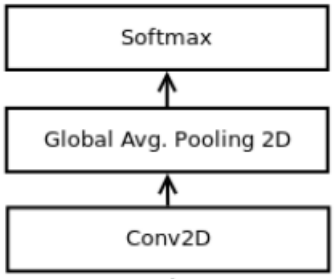
```
x = layers.add([x, residual])
```

این پروسه باید سه باردیگر تکرار شود که به صورت زیر می‌باشد:

```
 #module 2
residual = Conv2D(32, (1, 1), strides=(2, 2), padding='same', use_bias=False)(x)
residual = BatchNormalization()(residual)
```

```
x = SeparableConv2D(32, (3, 3), padding='same',
kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)

x = SeparableConv2D(32, (3, 3), padding='same',
kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)

x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
#adding two parameters
x = layers.add([x, residual])

#module 3
residual = Conv2D(64, (1, 1), strides=(2, 2),padding='same', use_bias=False)(x)
residual = BatchNormalization()(residual)

x = SeparableConv2D(64, (3, 3),
padding='same',kernel_regularizer=regularization,use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)

x = SeparableConv2D(64, (3, 3),
padding='same',kernel_regularizer=regularization,use_bias=False)(x)
x = BatchNormalization()(x)

x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
#adding two parameters
x = layers.add([x, residual])

 #module 4
residual = Conv2D(128, (1, 1), strides=(2, 2),padding='same', use_bias=False)(x)
residual = BatchNormalization()(residual)

x = SeparableConv2D (128, (3, 3), padding='same',
kernel_regularizer=regularization, use_bias=False) (x)
```

```
x = BatchNormalization () (x)
x = Activation ('relu') (x)

x = SeparableConv2D (128, (3, 3), padding='same',
kernel_regularizer=regularization, use_bias=False) (x)
x = BatchNormalization () (x)

x = MaxPooling2D ((3, 3), strides= (2, 2), padding='same')(x)
#adding two parameters
x = layers.add ([x, residual])
```



شکل ۳

کد های زیر اعمال قسمتی از معماری شبکه است که در شکل ۳ نشان داده شده است:

```
x = Conv2D(num_classes, (3, 3), padding='same')(x)
# reduces each feature map into a scalar value to extract global features from the
input image
x = GlobalAveragePooling2D()(x)
output = Activation('softmax',name='predictions')(x)
```

در مرحله بعد می خواهیم مدل را بسازیم:

```
model = Model(img_input, output)
model.compile(optimizer='adam',
loss='categorical_crossentropy',metrics=['accuracy'])
```

مدل با بهینه‌ساز آدام ساخته شده‌است و loss را تفاوت خروجی با هدف و metrics را دقت مدل می-گذاریم. حالا اگر یک چکیده از مدل را نمایش دهیم به صورت زیر است.(شکل۴).

```
model.summary()
```

حال که از صحت عملکرد مدل اطمینان حاصل کردیم باید مدل نهایی را ذخیره کنیم تا هر لحظه بتوان به آن دست‌رسی آسان داشت. در مرحله‌ی بعد برای دقت بیشتر دیتاست را گسترش می دهیم با زوم کردن در تصاویر و یا چرخندن، شیفت دادن آن‌ها به چپ و راست دیتاست کامل‌تری برای ورودی مدل فراهم می کنیم.

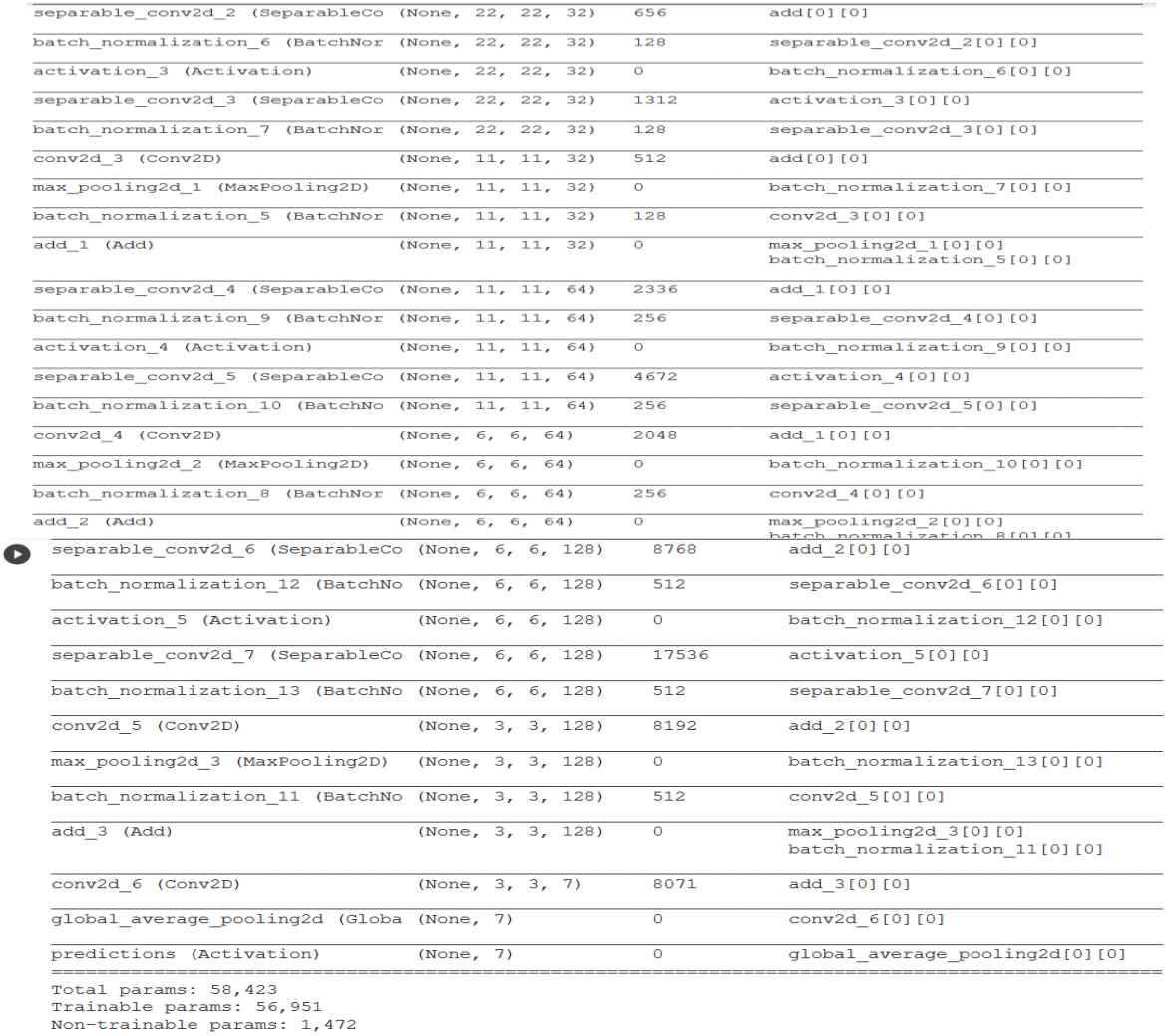
```
data_generator = ImageDataGenerator(
                featurewise_center=False,
                featurewise_std_normalization=False,
                rotation_range=10,
                width_shift_range=0.1,
                height_shift_range=0.1,
                zoom_range=.1,
                horizontal_flip=True)
```

و در انتها ورودی ساخته شده را به مدل خواهیم داد و مدل شروع به یادگیری می‌کند:

```
model.fit_generator(data_generator.flow(xtrain, ytrain,batch_size),
                steps_per_epoch=len(xtrain) / batch_size,
                epochs=num_epochs, verbose=1, callbacks=callbacks,
                validation_data=(xtest,ytest))
```

```
Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 48, 48, 1)]	0	
conv2d (Conv2D)	(None, 46, 46, 8)	72	input_1[0][0]
batch_normalization (BatchNorma	(None, 46, 46, 8)	32	conv2d[0][0]
activation (Activation)	(None, 46, 46, 8)	0	batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 44, 44, 8)	576	activation[0][0]
batch_normalization_1 (BatchNor	(None, 44, 44, 8)	32	conv2d_1[0][0]
activation_1 (Activation)	(None, 44, 44, 8)	0	batch_normalization_1[0][0]
separable_conv2d (SeparableConv	(None, 44, 44, 16)	200	activation_1[0][0]
batch_normalization_3 (BatchNor	(None, 44, 44, 16)	64	separable_conv2d[0][0]
activation_2 (Activation)	(None, 44, 44, 16)	0	batch_normalization_3[0][0]
separable_conv2d_1 (SeparableCo	(None, 44, 44, 16)	400	activation_2[0][0]
batch_normalization_4 (BatchNor	(None, 44, 44, 16)	64	separable_conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 22, 22, 16)	128	activation_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 22, 22, 16)	0	batch_normalization_4[0][0]
batch_normalization_2 (BatchNor	(None, 22, 22, 16)	64	conv2d_2[0][0]
add (Add)	(None, 22, 22, 16)	0	max_pooling2d[0][0] batch_normalization_2[0][0]



separable_conv2d_2 (SeparableCo	(None, 22, 22, 32)	656	add[0][0]	
batch_normalization_6 (BatchNor	(None, 22, 22, 32)	128	separable_conv2d_2[0][0]	
activation_3 (Activation)	(None, 22, 22, 32)	0	batch_normalization_6[0][0]	
separable_conv2d_3 (SeparableCo	(None, 22, 22, 32)	1312	activation_3[0][0]	
batch_normalization_7 (BatchNor	(None, 22, 22, 32)	128	separable_conv2d_3[0][0]	
conv2d_3 (Conv2D)	(None, 11, 11, 32)	512	add[0][0]	
max_pooling2d_1 (MaxPooling2D)	(None, 11, 11, 32)	0	batch_normalization_7[0][0]	
batch_normalization_5 (BatchNor	(None, 11, 11, 32)	128	conv2d_3[0][0]	
add_1 (Add)	(None, 11, 11, 32)	0	max_pooling2d_1[0][0] batch_normalization_5[0][0]	
separable_conv2d_4 (SeparableCo	(None, 11, 11, 64)	2336	add_1[0][0]	
batch_normalization_9 (BatchNor	(None, 11, 11, 64)	256	separable_conv2d_4[0][0]	
activation_4 (Activation)	(None, 11, 11, 64)	0	batch_normalization_9[0][0]	
separable_conv2d_5 (SeparableCo	(None, 11, 11, 64)	4672	activation_4[0][0]	
batch_normalization_10 (BatchNo	(None, 11, 11, 64)	256	separable_conv2d_5[0][0]	
conv2d_4 (Conv2D)	(None, 6, 6, 64)	2048	add_1[0][0]	
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0	batch_normalization_10[0][0]	
batch_normalization_8 (BatchNor	(None, 6, 6, 64)	256	conv2d_4[0][0]	
add_2 (Add)	(None, 6, 6, 64)	0	max_pooling2d_2[0][0] batch_normalization_8[0][0]	
separable_conv2d_6 (SeparableCo	(None, 6, 6, 128)	8768	add_2[0][0]	
batch_normalization_12 (BatchNo	(None, 6, 6, 128)	512	separable_conv2d_6[0][0]	
activation_5 (Activation)	(None, 6, 6, 128)	0	batch_normalization_12[0][0]	
separable_conv2d_7 (SeparableCo	(None, 6, 6, 128)	17536	activation_5[0][0]	
batch_normalization_13 (BatchNo	(None, 6, 6, 128)	512	separable_conv2d_7[0][0]	
conv2d_5 (Conv2D)	(None, 3, 3, 128)	8192	add_2[0][0]	
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 128)	0	batch_normalization_13[0][0]	
batch_normalization_11 (BatchNo	(None, 3, 3, 128)	512	conv2d_5[0][0]	
add_3 (Add)	(None, 3, 3, 128)	0	max_pooling2d_3[0][0] batch_normalization_11[0][0]	
conv2d_6 (Conv2D)	(None, 3, 3, 7)	8071	add_3[0][0]	
global_average_pooling2d (Globa	(None, 7)	0	conv2d_6[0][0]	
predictions (Activation)	(None, 7)	0	global_average_pooling2d[0][0]	

Total params: 58,423
Trainable params: 56,951
Non-trainable params: 1,472

شکل ۴

در این قسمت به نحوه‌ی استفاده از مدل آموزش‌دیده می‌پردازیم:

در google colab مدل را ساختیم حال برای استفاده از آن از pycharm استفاده می‌کنیم.

کتابخانه های مورد استفاده به شرح زیر است:

```
from keras.preprocessing.image import img_to_array
from keras.models import load_model
import imutils
import matplotlib.pyplot as plt
import cv2
import numpy as np
import sys
```

به این صورت که برای شناسایی چهره در تصاویر از مدل آماده هار که قبلا مطرح شده بود استفاده شده است و برای تشخیص حالت چهره از مدلی که ساخته‌ایم استفاده می‌کنیم در ادامه این دو مدل را import کرده‌ایم:

```
detection_model_path = 'haarcascade_frontalface_default.xml'   # Ready model face detection
emotion_model_path = '_mini_XCEPTION.96-0.64.hdf5'      #our method emotion recongnition
```

سپس در کد زیر اسم تصویری که می خواهیم توسط شبکه تست شود را قرار می دهیم.

```
'img_path = '1.jpg
```

حال مدل face detection و emotion detection را load می‌کنیم:

```
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry", "disgust", "scared", "happy", "sad", "surprised", "neutral"]
```

حال تصویر را خوانده وبه داده‌های مشخص‌شده می‌دهیم:

```
 #reading the frame
orig_frame = cv2.imread(img_path)
frame = cv2.imread(img_path, 0)
faces = face_detection.detectMultiScale(frame, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30)
flags=cv2.CASCADE_SCALE_IMAGE()
```

حال اگر صورتی در تصویر باشه کادری به ابعاد ۴۸*۴۸ را اطراف چهره انتخاب می‌شود آن را خاکستری کرده و در آرایه مشخص‌شده جای‌گذاری می‌شود.

```
if len(faces):
```

```
    faces = sorted(faces, reverse=True, key=lambda x: (x[2] - x[0]) * (x[3] -
x[1]))[0]
    (fX, fY, fW, fH) = faces
    roi = frame[fY:fY + fH, fX:fX + fW]
    roi = cv2.resize(roi, (48, 48))
    roi = roi.astype("float") / 255.0
    roi = img_to_array(roi)
    roi = np.expand_dims(roi, axis=0)
```

در این مرحله آرایه به مدل شبکه داده شد و احتمال هر یک از کلاس هارا بدست می آید و ماکسیمم احتمال در label قرار گرفت.

```
    preds = emotion_classifier.predict(roi)[0]
    emotion_probability = np.max(preds)
    label = EMOTIONS[preds.argmax()]
    cv2.putText(orig_frame, label, (fX, fY - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
    cv2.rectangle(orig_frame, (fX, fY), (fX + fW, fY + fH), (0, 0, 255), 2)
```

دو کد انتهایی برای قرار گیری یه مربع دور چهره و نوشتن label بالای کادر است. در ادامه همه احتمال‌ها برای هفت کلاس را نمایش می‌دهد و بیشترین احتمال را بیان می‌کند.

```
print('----------------------------Probabilistic results----------------------------')
print('angry:    ',preds[0])
print('disgust:  ',preds[1])
print('scared:   ',preds[2])
print('happy:    ',preds[3])
print('sad:      ',preds[4])
print('surprised:',preds[5])
print('neutral:  ',preds[6])
print('----------------------------Final Decision----------------------------')
```

```
print('Most likely this person is:',label)
```

Abstract

Understanding customers' emotions for jobs and businesses is provided by using updated and new technologies in this project. These technologies can be applied the sellig process at the best moment according to the consumer sentiment analysis. Emotional intelligence is a growing knowledge that has influence in notonly commercial purposes butalso in new start - ups, health care, wearable digital tools, human-robot communication, education, and etc with a great impact.

Increasing importance in accuracy has led us to the use of computer - aided diagnosis systems. In this project, we try to study power of deep learning to recognize facial expressions and improve the performance of the classifiers. For this purpose, the common and available dataset which is named FER-2013 is used.

In this project, we applied the convolution neural network (cnn) for real-time system design. Then we validate our model to perform facial recognition tasks and emotion classification by accuracy. The results show the improvment of emotion recognition in terms of accuracy increases to 71%.

Key Words: Emotions, Face Detection, Deep Learning, Convolutional Neural Network (CNN).

Amirkabir University of Technology
(Tehran Polytechnic)

Electrical Engineering

Bachelor Thesis

Emotion recognition using deep learning

By
Ramtin Asgarianamiri

Advisor
Dr. F. Abodlahi

September 2021