
# طبقه بندی تصاویر خنده دار اینترنتی بر اساس مفهوم کلی آنها (مثبت, منفی و یا خنثی) با استفاده شبکه ی پیش تعلیم داده شده ی Inception, پردازش تصویر و پردازش متن

پایان نامه براي دريافت درجه کارشناسي

در رشته مهندسي کامپيوتر - گرايش نرم‌افزار

نام دانشجو:

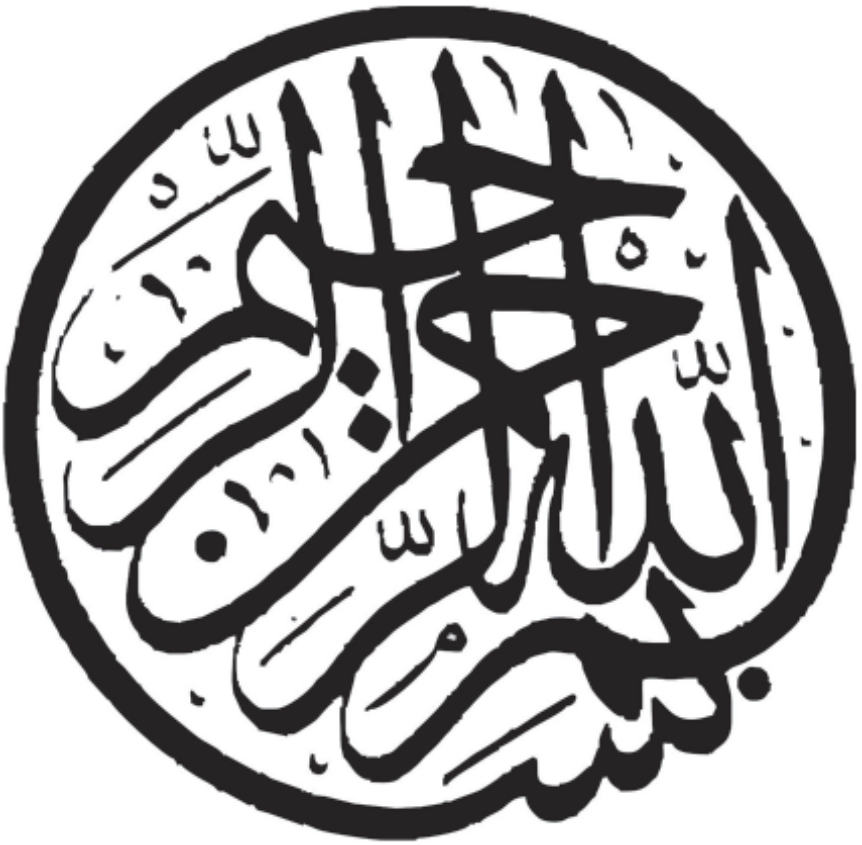
آرین شریعت

استاد راهنما:

دکتر مرتضی آنالویی

شهريور ماه ۱۴۰۰

دانشکده مهندسي کامپيوتر



# تأییدیه هیئت داوران جلسه دفاع از پایان‌نامه

نام دانشکده: مهندسي كامپيوتر

نام دانشجو: آرین شریعت

عنوان پایان‌نامه: طبقه بندی تصاویر خنده دار اینترنتی بر اساس مفهوم کلی آنها (مثبت, منفی و یا خنثی) با استفاده شبکه های از پیش تعلیم داده شده, پردازش تصویر و پردازش متن

تاریخ دفاع: شهریور ماه ۱۴۰۰

رشته: مهندسي كامپيوتر

گرایش: نرم‌افزار

| ردیف | سمت | نام و نام خانوادگي | مرتبه دانشگاهي | دانشگاه یا مؤسسه | امضاء |
|---|---|---|---|---|---|
| 1 | استاد راهنما | دکتر مرتضی آنالویی | استادیار | دانشگاه علم و صنعت ایران | |
| 2 | استاد مدعو خارجي | | استادیار | | |
| 3 | استاد مدعو داخلي | | | | |

# تأییدیه صحت و اصالت نتایج

## باسمه تعالي

اینجانب آرین شریعت باروق به شماره دانشجویی ۹۵۵۲۱۲۲۵ دانشجوی رشته مهندسی کامپیوتر گرایش نرم‌افزار مقطع تحصیلی کارشناسی ارشد تایید می نمایم که کلیه نتایج مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی عضو هیات علمی دانشگاه علم و صنعت ایران بدون هرگونه دخل و تصرف انجام گرفته و به موارد نسخه برداری شده از آثار دیگران، مطابق مقررات و ضوابط، ارجاع داده شده و مشخصات کامل منابع را در فهرست منابع ذکر کرده ام. این پایان‌نامه قبلاً برای احراز هیچ مدرکی ارائه نگردیده است.

در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم ( قانون حمایت از حقوق مؤلفان و منصفان و قانون ترجمه، تکثیر و نشریات و آثار صوتی، ضوابط و مقررات آموزشی و پژوهشی، انضباطی و غیره) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض در خصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می نمایم. در ضمن، مسئولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح ( اعم از اداری و قضایی) به عهده اینجانب خواهد بود و دانشگاه هیچ گونه مسئولیتی در این خصوص نخواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه علم و صنعت ایران است. هرگونه استفاده از نتایج علمی و عملی و واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه برداری ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه علم و صنعت ایران ممنوع است. نقل مطالب با ذکر منبع بلامانع است.

نام و نام خانوادگي: آرین شریعت

امضا و تاریخ:

# مجوز بهره‌برداري از پايان‌نامه

بهره‌برداري از اين پايان‌نامه در چارچوب مقررات كتابخانه و با توجه به محدوديتي كه توسط استاد راهنما به شرح زير تعيين مي‌شود، بلامانع است:

بهره‌برداري از اين پايان‌نامه براي همگان بلامانع است.

بهره‌برداري از اين پايان‌نامه با اخذ مجوز از استاد راهنما، بلامانع است.

بهره‌برداري از اين پايان‌نامه تا تاريخ ..... ممنوع است.

نام استاد:

تاريخ:

امضا:

# تقدیر و تشکر

از استاد گران‌قدر، جناب آقای دکتر مرتضی آنالویی که در طول مدت تحقیق، مرا از رهنمودها و تجارب با ارزش خویش بهره‌مند ساختند، صمیمانه سپاسگزارم.

## چکیده

پیشرفت های اخیر در زمینه ی اینترنت و استفاده سرشار از دنیای مجازی منجر به ظاهر شدن پلتفرم های سریع و بهینه در زمینه ارتباطات شده است. در این پلتفرم ها داده ها به صورت تصویر، متن و گفتار در جریان هستند. این مدیم ها سبب پدید آمدن پدیده ای منحصر به فرد و جدید به اسم **Internet memes** شده اند که آن را می توان به گونه ای تصاویر و مفاهیم خنده دار اینترنتی قلمداد کرد. این مفاهیم در اکثر اوقات به صورت تصاویری شوخ انگیز با متن های طعنه آمیز میباشند. بنابراین تحقیق دقیق و موثر بر روی این داده ها نیازمند روشی ترکیبی بوده تا همزمان تصویر و متن آنها تحلیل شوند. بررسی رویدادهای رخ داده در شبکه های اجتماعی به باور بیشتر صاحبنظران، از اهمیت بسیار والایی برخوردار است. علت این مساله را میتوان در سودمندی و در عین حال قدرت تخریب بالای این شبکه ها دانست؛ بخصوص با درنظر گرفتن تاثیر بالایی که بر گروههای سنی پایین جوامع دارند. در این پروژه هدف کلاس بندی کردن این تصاویر به مفاهیم کلی مثبت، منفی و خنثی میباشد. با استفاده از این مدل میتوان تصاویر توهین آمیز در فضای مجازی را شناسایی کرد. برای رسیدن به این هدف از شبکه عمیق یادگیری شده **Inception** برای تصاویر و **LSTM** برای متن ها استفاده شده است.



**واژه‌هاي كليدي:** یادگیری ماشین، شبکه عمیق، یادگیری عمیق، شبکه اجتماعی

# فهرست مطالب

## فهرست شکل‌ها

# فصل 1:

# مقدمه

## 1-1- شرح مسئله

با در نظر گرفتن تعداد بسیار زیاد تصاویری که این روز ها در پلتفرم های مجازی مانند: facebook و twitter و instagram و ... منتشر میشوند گوناگونی مسائلی که مربوط به فهم تصاویر و نوشته ها با سرعتی شگفت آور رو به فزونی است. حل این مسائل نیاز به متد های دگرگونی دارد که در راه حل های مطرح شده گذشته نمیگنجد. چرا که راه حل هایی که در گذشته مطرح شده اند دیدی از خود تصویر جدای متن ندارند.

هدف ما در این پروژه تحلیل تصاویر خنده دار اینترنتی و استخراج حس کلی این تصاویر میباشد که میتواند حسی مثبت منفی و یا خنثی باشد.

لازم به ذکر است که پروژه و پیاده سازی ذکر شده از تسک های مسابقه هوش مصنوعی SEMEVAL2020 تحت عنوان Memotion Analysis میباشد.

# فصل 2:

# داده ها

## 2-1- مقدمه

استخراج متن از تصویر اهمیتی زیادی در کنار تحلیل خود تصویر دارد. به همین دلیل کار های مرتبط در دو بخش بررسی میشوند که اولی مربوط به استخراج متن از تصاویر و دومی تحلیل و آنالیز تصاویر میباشد.

## 2-2- دیتاست

مسابقه به این منظور ۶۹۸۸ تصویر دارای متن از ۵۲ کتگوری مختلف مهیا کرده است. متن های تصاویر نیز به صورت جداگانه به کمک OCR استخراج شده اند و در دسترس شرکت کنندگان قرار گرفته است. لازم به ذکر است دیتا در قلب فایل اکسل منتشر شده که شامل متن تصویر و مفهوم کلی تصویر و url تصویر مورد نظر میباشد. تصاویر توسط API GOOGLE دانلود و در درایو شخصی ذخیره شده اند.

## 2-3- فیلتر داده ها

برای جلوگیری از پردازش داده های پرت و نویزی داده ها از فیلتر های زیر عبور کرده اند:

- تصویر باید پشت زمینه خالی داشته و همچنین داری متن تعبیه شده باشد.
- برای این تحقیق فقط از تصاویر با متون انگلیسی استفاده شده است.

## 2-4- پیش پردازش داده ها

- پاک کردن سطر هایی که متنشان استخراج نشده است.
- تمامی متن ها lowercase شده اند.
- پاک کردن کلمات معمولی انگلیسی مانند am is are
- پاک کردن URL عکس ها درمتن استخراج شده
- پاک کردن نام کاربری کاربران از تصاویر (به ویژه توییتر)
- پاک کردن تمامی کاراکتر هایی که جزو الفبای انگلیسی نیستند

## 2-5- داده یادگیری و تست

به صورت تصادفی ۶۲۸۹ تصویر برای یادگیری و مابقی برای تست انتخاب شده اند.

# فصل 3:

# مروري بر كارهاي مرتبط

## 3-1- مقدمه

در فصل قبل به مفاهیم پایه و تعاریف لازم مرتبط با موضوع پژوهش پرداخته شد. در این فصل با مروری بر کارهای مرتبط، به معرفی برخی از مدل‌هایی که به استخراج متن از تصویر و تحلیل تصویر پرداخته اند میپردازیم.

## 3-2- بررسی کارهای انجام شده در حوزه آنالیز متن و تصویر

- [25] جزو اولین روش های استخراج متن از تصویر و تبدیل آنها به مجموعه های کاراکتری ثابت به کمک CNN میباشد.
- [26] متن استخراج شده به کمک OCR را به کمک یک مدل n-gram تصحیح میکند.
- [27] فعالیت هایی در حوزه استخراج کاراکتر های دست نوشته
- [28] روش های مختلف استخراج متن به کمک OCR
- [29] جزو اولین کار ها در زمینه استخراج توهین در متن ها در فضای مجازی
- [3] جزو اولین کار ها در زمینه استخراج توهین در تصاویر در فضای مجازی

| Team | Inception | ResNet | BERT | XLNet | LSTM | GRU | CNN | VGG-16 | DenseNet | GloVe |
|---|---|---|---|---|---|---|---|---|---|---|
| Hitachi | ✓ | ✓ | ✓ | ✓ | | | | | | |
| YNU-HPCC | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | |
| PRHLT-UPV | | | ✓ | | ✓ | | ✓ | ✓ | ✓ | |
| Guoym | | ✓ | ✓ | | | ✓ | | | | |
| Vkeswani IITK | | ✓ | ✓ | | | | | | | |
| Memebusters | ✓ | | ✓ | | ✓ | ✓ | | | | ✓ |
| Sunil Gundapu | ✓ | | | | | ✓ | ✓ | | | ✓ |
| Suciati Indra | | | | | | | | ✓ | | |
| SESAM Bonheme | | | | | | | | | | |
| Zehao Liu | | | | | ✓ | | ✓ | | | |
| NUAA-QMUL | | ✓ | ✓ | | | | | | ✓ | |
| Ambuje Gupta | | | | | | | | ✓ | | |
| CN-HIT-MLT | | ✓ | ✓ | | | | | | | |
| KAFK | | | ✓ | | | | | | | |
| NIT-Agartala-NLP-Team | | | ✓ | | ✓ | | | | | ✓ |
| DSC IIT-ISM | | ✓ | | | ✓ | | | | | |
| Sabino Infotech | ✓ | ✓ | | | | | | | | |
| UPB | | | ✓ | | | | | ✓ | | |
| Sravani IS | | | | | ✓ | | ✓ | | | |
| NAYEL | | | | | | | | | | |
| IIITG-ADBU | ✓ | | ✓ | | ✓ | | | ✓ | | ✓ |
| LT3 | | ✓ | | | | | | | | |
| Urszula | | | | | ✓ | | ✓ | ✓ | | ✓ |
| CSECU KDE MA | | | | | ✓ | | | | | |
| Ingroj Jonathan | | | | | ✓ | ✓ | | | | |
| Adithya Sanath | | | ✓ | ✓ | ✓ | | | | | |

شکل (3-1) چک لیستی از مدل های از پیش تعلیم داده شده

# فصل 4:

# روش پيشنهادي

## 4-1- مقدمه

به منظور دستیابی به هدف مورد نظر که فصول پیش توضیح داده شد ابتدا باید چالش های پیش رو را شناخت و سپس روشی براین اساس انتخاب کرد. یکی از چالش های اصلی تحلیل تصاویر خنده دار اینترنتی دشوار بودن تشخیص بار مفهومی یک مطلب طنز میباشد که حتی برای انسان نیز پیچیدگی هایی دارد. چالش دیگر دشوار بودن تحلیل این عکس هاست چرا که تصویر و متن در هم آمیخته شده اند و در کنار یکدیگر مفهوم گرفته اند.

روش پیشنهادی اعمال شده برای ساخت مدل کارا استفاده از embedding و شبکه بازگشتی RNN برای متن ها و فاین تیون کردن تصاویر به کمک مدل از پیش پروسس شده Inception میباشد. سپس این دو شاخه با هم ترکیب شده و به منظور تعیین خروجی مسئله به یک شبکه MLP داده میشود.

## 4-2- مفاهیم پایه

در این بخش به مفاهیم پایه شبکه های عصبی و تنظیم دقیق (Fine Tune) مدل می پردازیم.

## 4-3- شبکه عصبی مصنوعی

یک شبکهٔ عصبی مصنوعی، از سه لایهٔ ورودی، خروجی و پردازش تشکیل می‌شود. هر لایه شامل گروهی از سلول‌های عصبی (نورون) است که عموماً با کلیهٔ نورون‌های لایه‌های دیگر در ارتباط هستند، مگر این که کاربر ارتباط بین نورون‌ها را محدود کند؛ ولی نورون‌های هر لایه با سایر نورون‌های همان لایه، ارتباطی ندارند.

نورون کوچک‌ترین واحد پردازشگر اطلاعات است که اساس عملکرد شبکه‌های عصبی را تشکیل می‌دهد. یک شبکهٔ عصبی مجموعه‌ای از نورون‌هاست که با قرار گرفتن در لایه‌های مختلف، معماری خاصی را بر مبنای ارتباطات بین نورون‌ها در لایه‌های مختلف تشکیل می‌دهند. نورون می‌تواند یک تابع ریاضی غیرخطی باشد، در نتیجه یک شبکهٔ عصبی که از اجتماع این نورون‌ها تشکیل می‌شود، نیز می‌تواند یک سامانهٔ کاملاً پیچیده و غیرخطی باشد. در شبکهٔ عصبی هر نورون به‌طور مستقل عمل می کند و رفتار کلی شبکه، برآیند رفتار نورون‌های متعدد است. به عبارت دیگر، نورون‌ها در یک روند همکاری، یکدیگر را تصحیح می کنند.

یادگیری ماشینی با نظارت (supervised learning) به دنبال تابعی از میان یک سری توابع هست که تابع هزینه (loss function) داده‌ها را بهینه سازد. به عنوان مثال در مسئله رگرسیون تابع هزینه می‌تواند اختلاف بین پیش‌بینی

و مقدار واقعی خروجی به توان دو باشد، یا در مسئله طبقه‌بندی ضرر منفی لگاریتم احتمال خروجی باشد. مشکلی که در یادگیری شبکه‌های عصبی وجود دارد این است که این مسئله بهینه‌سازی دیگر محدب (convex) نیست.ازین رو با مشکل کمینه‌های محلی روبرو هستیم. یکی از روش‌های متداول حل مسئله بهینه‌سازی در شبکه‌های عصبی بازگشت به عقب یا همان back propagation است. روش بازگشت به عقب گرادیانِ تابع هزینه را برای تمام وزن‌های شبکه عصبی محاسبه می‌کند و بعد از روش‌های گرادیان کاهشی (gradient descent) برای پیدا کردن مجموعه وزن‌های بهینه استفاده می‌کند. روش‌های گرادیان کاهشی سعی می‌کنند به صورت متناوب در خلاف جهت گرادیان حرکت کنند و با این کار تابع هزینه را به حداقل برسانند. پیدا کردن گرادیانِ لایه آخر ساده است و با استفاده از مشتق جزئی بدست می‌آید. گرادیانِ لایه‌های میانی اما به صورت مستقیم بدست نمی‌آید و باید از روش‌هایی مانند قاعده زنجیری در مشتق‌گیری استفاده کرد. روش بازگشت به عقب از قاعده زنجیری برای محاسبه گرادیان‌ها استفاده می‌کند و همان‌طور که در پایین خواهیم دید، این روش به صورت متناوب گرادیان‌ها را از بالاترین لایه شروع کرده آن‌ها را در لایه‌های پایینتر «پخش» می‌کند.

## 4-4- پرسپترون چندلایه (MLP)

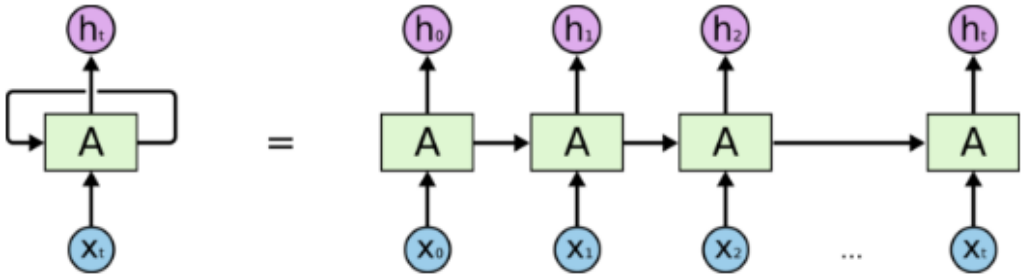
پرسپترون چند لایه، (به انگلیسی: Multilayer perceptron) دسته ای از شبکه‌های عصبی مصنوعی پیشخور است. یک MLP شامل حداقل سه لایه گره است: یک لایه ورودی، یک لایه پنهان و یک لایه خروجی. به جز گره‌های ورودی، هر گره یک نورون است که از یک تابع فعال‌سازی غیر خطی استفاده می‌کند. MLP از تکنیک یادگیری نظارت شده به نام بازپرداخت برای آموزش استفاده می‌کند. لایه‌های متعدد آن و فعال‌سازی غیرخطی آن MLP را از یک پرسپترون خطی متمایز می‌کند. در واقع می‌تواند داده‌هایی را متمایز کند که به صورت خطی قابل‌تفکیک نیستند.

اگر یک پرسپترون چند لایه، تابع فعال‌سازی خطی در تمام نورون‌ها داشته باشد، در واقع با این تابع خطی ورودی‌های وزن دار هر نورون را ترسیم می‌کند. سپس با استفاده از جبر خطی نشان می‌دهد که هر عددی مربوط به لایه‌ها را می‌توان به یک مدل ورودی - خروجی دو لایه کاهش داد. در MLP، برخی از نورون‌ها از یک تابع فعال غیرخطی استفاده می‌کنند که برای مدل‌سازی فرکانس پتانسیل‌های عمل یا شلیک نورون‌های بیولوژیکی توسعه داده شده‌است.

MLP شامل سه یا تعداد بیشتری از لایه‌ها است که از گره‌های غیرخطی فعال کننده هستند. از آنجا که MLPها به‌طور کامل متصل شده‌اند، هر گره در یک لایه با وزن مشخص در هر نود در لایه بعدی متصل می‌شود.

## 4-5- شبکه عصبی بازگشتی (Recurrent Neural Networks)

شبکه عصبی بازگشتی (RNN) که به آن شبکه عصبی مکرر نیز گفته می شود، نوعی از شبکه عصبی مصنوعی است که در تشخیص گفتار، پردازش زبان طبیعی (NLP) و همچنین در پردازش داده های ترتیبی (Sequential data) استفاده می شود. بسیاری از شبکه های عمیق مانند CNN شبکه های پیش خور (Feed Forward) هستند یعنی سیگنال در این شبکه ها فقط در یک جهت از لایه ورودی، به لایه های مخفی و سپس به لایه خروجی حرکت می کند و داده های قبلی به حافظه سپرده نمی شوند. اما شبکه های عصبی بازگشتی (RNN) یک لایه بازخورد دارند که در آن خروجی شبکه به همراه ورودی بعدی، به شبکه بازگرداننده می شود. RNN می تواند به علت داشتن حافظه داخلی، ورودی قبلی خود را به خاطر بسپارد و از این حافظه برای پردازش دنباله ایی از ورودی ها استفاده کند. به بیان ساده، شبکه‌های عصبی بازگشتی شامل یک حلقه بازگشتی هستند که موجب می شود اطلاعاتی را که از لحظات قبلی بدست آورده ایم از بین نروند و در شبکه باقی بمانند.




An unrolled recurrent neural network.

شکل (4-1) تصویری از شبکه بازگشتی

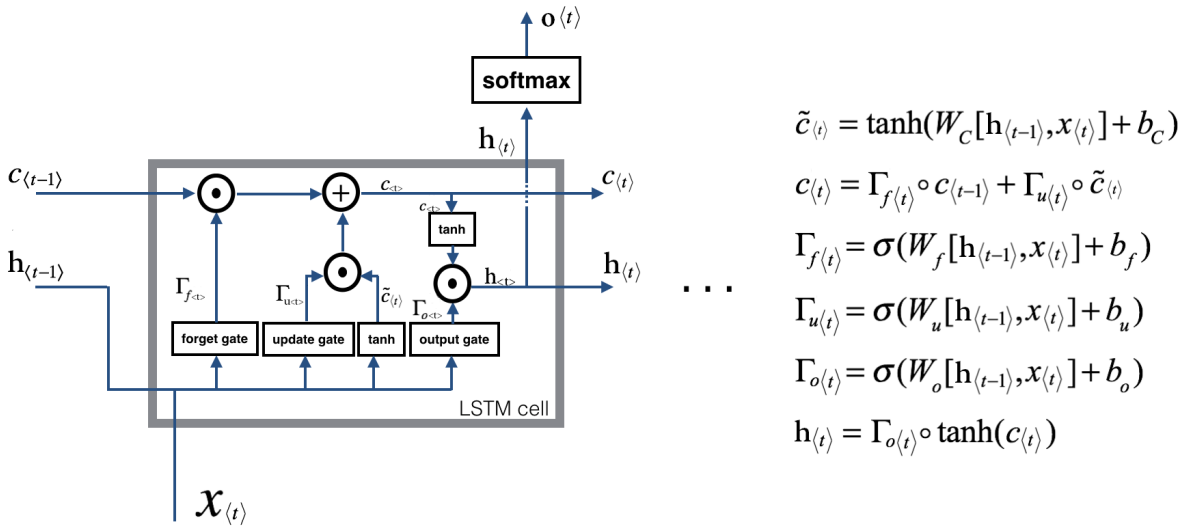
## 4-6- حافظه طولانی کوتاه-مدت (LSTM)

برخلاف شبکه عصبی بازگشتی سنتی که صرفا جمع متوازن سیگنالهای ورودی را محاسبه کرده و سپس از یک تابع فعالسازی عبور میدهد هر واحد LSTM از یک حافظه $C_t$ در زمان $t$ بهره میبرد. خروجی $h_t$ و یا فعالسازی واحد LSTM بصورت $h_t = \Gamma_o.tanh(C_t)$ است که در آن $\Gamma_o$ دروازه خروجی است که کنترل کننده میزان محتوایی است که از طریق حافظه ارائه میشود. دروازه خروجی از طریق عبارت $\Gamma_o = \sigma(W_o.[h_{t-1}, X_t] + b_o)$ محاسبه میشود که در آن $\sigma$ تابع فعال سازی سیگموید است. $W_o$ نیز یک ماتریس اریب است. سلول حافظه $C_t$ نیز با فراموشی نسبی حافظه فعلی و اضافه کردن محتوای حافظه جدید بصورت $\hat{C}_t$ بروز رسانی میشود که در آن محتوای حافظه جدید از طریق عبارت $C_t = \Gamma_f.C_{t-1} + \Gamma_u.\hat{C}_t$ بدست می آید. آن میزان از حافظه فعلی که باید فراموش شود توسط دروازه فراموشی $\Gamma_f$ کنترل میشود و آن میزانی از محتوای حافظه جدید که باید به سلول حافظه اضافه شود توسط دروازه بروزرسانی (یا بعضا به دروازه ورودی معروف است) انجام میگیرد. این عمل با محاسبات زیر صورت میگیرد :

$$\hat{C}_t = tanh(W_C.[h_{t-1}, X_t] + b_c)$$

$$\Gamma_f = \sigma(W_f.[h_{t-1}, X_t] + b_f)$$

$$\Gamma_u = \sigma(W_u.[h_{t-1}, X_t] + b_u)$$




$$\tilde{c}_{\langle t\rangle} = \tanh(W_C[\mathrm{h}_{\langle t-1\rangle}, x_{\langle t\rangle}] + b_C)$$
$$c_{\langle t\rangle} = \Gamma_{f\langle t\rangle} \circ c_{\langle t-1\rangle} + \Gamma_{u\langle t\rangle} \circ \tilde{c}_{\langle t\rangle}$$
$$\Gamma_{f\langle t\rangle} = \sigma(W_f[\mathrm{h}_{\langle t-1\rangle}, x_{\langle t\rangle}] + b_f)$$
$$\Gamma_{u\langle t\rangle} = \sigma(W_u[\mathrm{h}_{\langle t-1\rangle}, x_{\langle t\rangle}] + b_u)$$
$$\Gamma_{o\langle t\rangle} = \sigma(W_o[\mathrm{h}_{\langle t-1\rangle}, x_{\langle t\rangle}] + b_o)$$
$$\mathrm{h}_{\langle t\rangle} = \Gamma_{o\langle t\rangle} \circ \tanh(c_{\langle t\rangle})$$

شکل (2-4) شبکه عصبی بازگشتی LSTM

در شبکه عصبی LSTM ما با مفاهیم جدیدی مواجه میشویم که در شبکه عصبی بازگشتی سنتی وجود نداشتند. در این شبکه اصطلاحا سه دروازه یا gate وجود دارد که از طریق آن شبکه نسبت به کنترل جریان داده درون خود اقدام میکند.

این سه دروازه عبارتند از :

دروازه نسیان یا فراموشی (Forget gate)

دروازه بروزرسانی (Update gate) (به دروازه ورودی یا Input gate هم معروف است)

و دروازه خروجی (Output gate)

دروازه فراموشی یا همان Forget gate که در عبارات بالا بصورت $\Gamma_f$ نمایش داده شده است، وظیفه کنترل جریان اطلاعات از گام زمانی قبلی را دارد. این دروازه مشخص میکندآیا اطلاعات حافظه از گام زمانی قبل مورد استفاده قرار گیرد یا خیر و اگر بایداز گام زمانی قبل چیزی وارد شود به چه میزان باشد.

دروازه بروزرسانی یا همان Update gate که در عبارات بالا بصورت $\Gamma_u$ نمایش داده شده است، وظیفه کنترل جریان اطلاعات جدید را بر عهده دارد. این دروازه مشخص میکند آیا در گام زمانی فعلی باید از اطلاعات جدید مورد استفاده قرار گیرد یا خیر و اگر بلی به چه میزان. از این دروازه عموما به دروازه ورودی نیز یاد میشود.

دروازه خروجی یا همان Output gate که در عبارات بالا بصورت $\Gamma_o$ نمایش داده شده است، نیز مشخص میکند چه میزان از اطلاعات گام زمانی قبل با اطلاعات گام زمانی فعلی به گام زمانی بعد منتقل شود.

وجود این دروازه ها به این شکل است که مکانیزم کنترلی بسیار دقیقی را ایجاد میکند. حالا اجازه دهید با یک مثال کمی این مفاهیم را واضح تر بیان کنیم :

## دروازه فراموشی :

شکل (4-3) دروازه فراموشی در LSTM

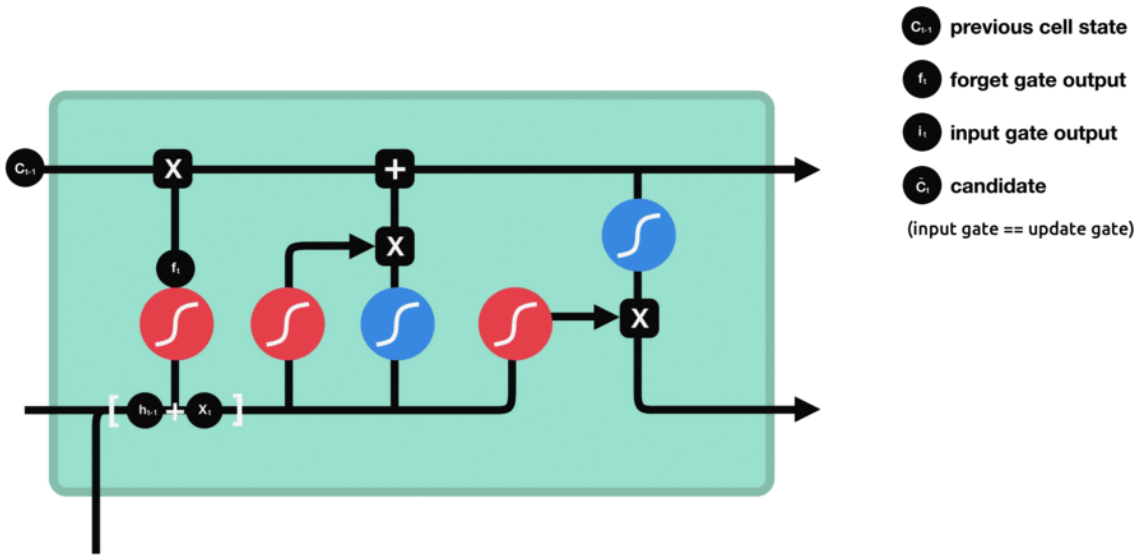
برای چک و کنترل LSTM فرض کنید ما چند کلمه از یک متن را از ورودی می خوانیم و می خواهیم از یک ساختار
گرامر استفاده کنیم (مثلا ببینیم آیا فاعل مفرد است یا جمع ). اگر فاعل از مفرد به جمع تغییر پیدا کرد(یا بالعکس)

این کار از LSTM ما باید راهی پیدا کنیم تا مقدار ذخیره شده قبلی در حافظه را با حالت جدید تعویض کنیم. در طریق دروازه forget به صورت زیر انجام میشود:

$$\Gamma_f = \sigma(W_f.[h_{t-1}, X_t] + b_f)$$

در اینجا $W_f$ ماتریس وزنی است که رفتار دروازه فراموشی را کنترل میکند. در بخش قبل دیدیم که برای سادگی کار چطور بردارهای $X_t$ و $h_{t-1}$ را با هم ترکیب میکنیم و در یک عملیات آنها را شرکت می دهیم. (برای مرور اینجا را ببینید). اگر ما عملیات فوق را انجام دهیم چون از تابع فعال سازی سیگموید استفاده میکنیم نتیجه برداری بنام $\Gamma_f$ خواهد بود که مقادیری بین ۰ و ۱ خواهد داشت. این بردار سپس در عبارت بعدی در $C_{t-1}$ ضرب خواهد شد. بنابراین اگر مقادیر بردار دروازه فراموشی $\Gamma_f$ صفر باشد(یا به سمت صفر میل کند) عملا به معنای در نظر نگرفتن محتوای $C_{t-1}$ است. به عبارت ساده تر یعنی شبکه اطلاعات ارائه شده توسط $C_{t-1}$ را دور انداخته و هیچ توجهی به آن نمیکند. به همین صورت اگر مقادیر بردار $\Gamma_f$ ۱ باشد این اطلاعات توسط شبکه حفظ میشود.مقادیر مابینی نیز موجب میشود شبکه به همان میزان از محتوای ارائه شده از گام زمانی قبل استفاده کند(یعنی بخشی را دور ریخته و از بخش دیگر استفاده کند).

دروازه بروزرسانی:




شکل (4-4) دروازه بروزرسانی در LSTM

حالا بعد از اینکه با موفقیت فراموش کردیم که فاعل ما مفرد است(یعنی مقادیر قبلی حافظه که اشاره به مفرد بودن فاعل
داشت را پاک کردیم)، نیاز داریم تا راهی پیدا کنیم تا نشان دهیم که الان فاعل جمع است (و دیگر مفرد نیست) (یعنی در ورودی ما با فاعل جمع سرو کار داریم (داده الان ما فاعلش جمع است!). اینجا از دروازه بروزرسانی استفاده می کنیم که به صورت زیر محاسبه میشود

$$\Gamma_u = \sigma(W_u.[h_{t-1}, X_t] + b_u)$$

حالا برای بروزرسانی فاعل جدید، ما نیاز به یک بردار جدید داریم که بتوانیم آنرا با حالت قبلی حافظه جمع کنیم پس : برای اینکار بصورت زیر عمل میکنیم. ابتدا بردار جدیدی که صحبتش را کردیم بصورت زیر ایجاد میکنیم

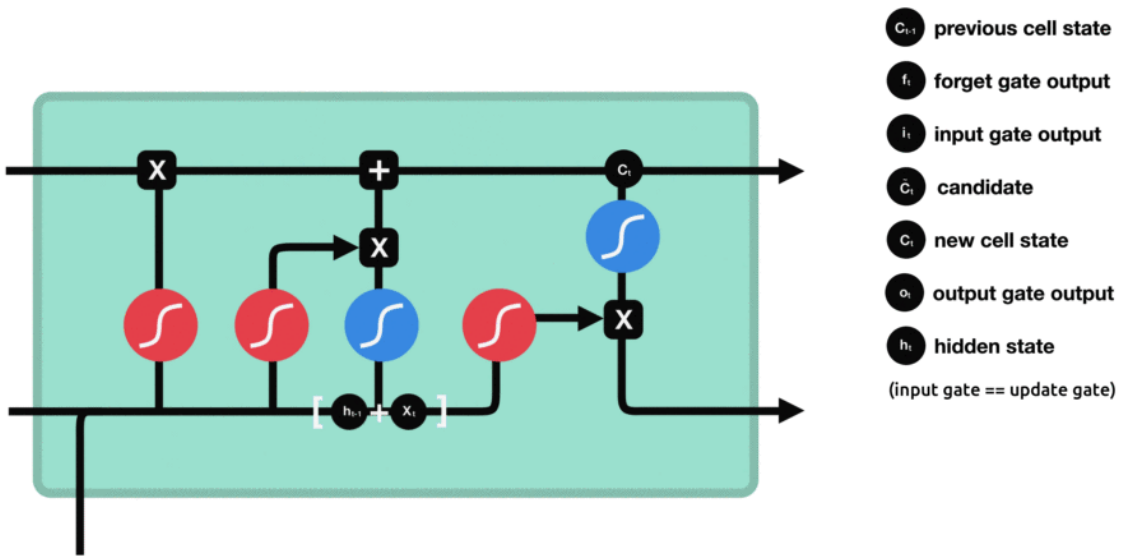
$$\hat{C}_t = tanh(W_C.[h_{t-1}, X_t] + b_c)$$

: و در آخر هم حافظه را بروز رسانی میکنیم

$$C_t = \Gamma_f.C_{t-1} + \Gamma_u.\hat{C}_t$$

در عبارت فوق بخش ابتدایی مشخص کننده این است که چه میزان اطلاعات از بخش قبل(حافظه از گام زمانی قبل) استفاده شود و بخش دوم حاوی اطلاعات جدید است که مورد استفاده قرار میگیرد.

: دروازه خروجی




شکل (4-5) دروازه خروجی در LSTM

در انتها نیز برای اینکه مشخص کنیم در خروجی از چه محتوایی باید استفاده کنیم از دروازه خروجی بهره میبریم.

: شیوه کار بصورت زیر است

$$\Gamma_o = \sigma(W_o.[h_{t-1}, X_t] + b_o)$$

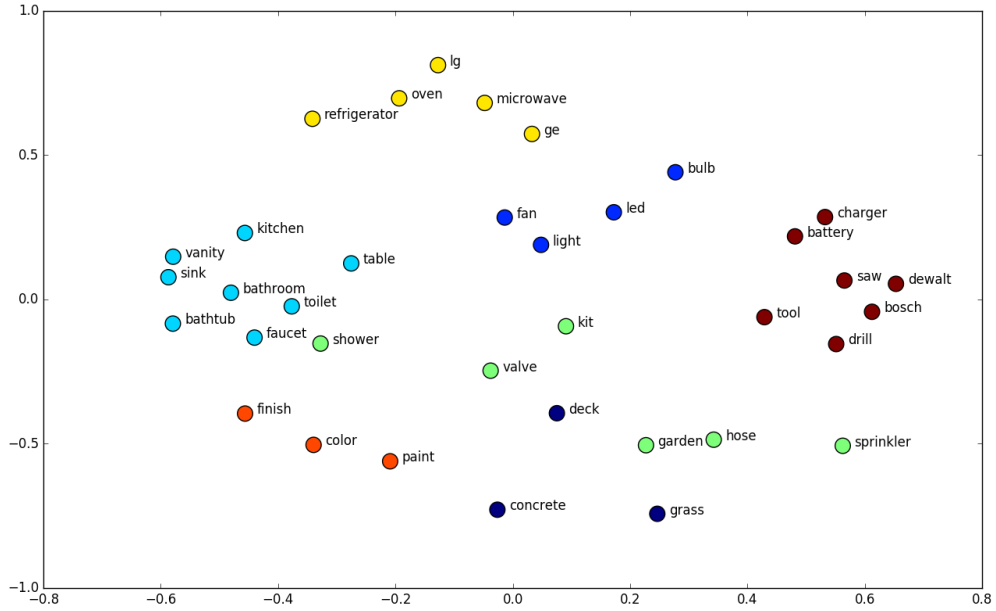
$$h_t = \Gamma_o.tanh(C_t)$$

## 4-7- تعبیه سازی کلمات (Word Embedding)

مفهوم اصلی word embedding تمامی لغات استفاده شده در یک زبان را میتوان توسط مجموعه ای از اعداد اعشاری (در قالب یک بردار) بیان کرد. Word embedding ها بردارهای n-بعدی ای هستند که تلاش میکنند معنای لغات و محتوای آنها را با مقادیر عددی خود ثبت و ضبط کنند. هر مجموعه ای از اعداد یک "بردار کلمه" معتبر به حساب می آید که الزاما برای ما سودمند نیست، آن مجموعه ای از بردار کلمات برای کاربردهای مورد نظر ما سودمندند که معنای کلمات ، ارتباط بین آنها و محتوای کلمات مختلف را همانطور که بصورت طبیعی [توسط ما] مورد استفاده قرار گرفته اند، بدست آورده باشند.

در فضای word embedding کلمات مشابه به مکان های مشابهی در فضای N-D بعدی همگرا میشوند. در تصویر پایین کلمه "مایکروویو"، "گاز" و "یخچال" هر سه در مکان مشابهی در فضای embedding قرار میگیرند ، بسیار دورتر از مکان کلمات بی ربطی مثل "چمن"، "باغچه" ، "بتن" و...

شباهتی که در اینجا از آن صحبت میکنیم را را میتوان توسط فاصله اقلیدسی (فاصله واقعی بین نقاط در فضای N-D بُعدی) و یا شباهت کسینوسی یا اصطلاحا Cosine Similarity (زاویه بین دو بردار در فضای برداری) تعریف نمود.




شکل (۴-۶) مثالی از یک فضای دو بعدی word embedding که در آن کلمات مشابه در مکان های مشابهی یافت میشوند.

## 4-8- تنظیم دقیق (Fine tuning)

هدف از fine tuning تنظیم مدل های موجود و از پیش پروسس شده جهت رسیدن به مدل جدید مناسب برای داده های ما میباشد. در این روش دیگر نیاز به داده های حجیم و زمان آموزش طولانی نیست و به راحتی با داده های کم میتوان در زمانی کوتاه به مدل مطلوب رسید. نکته ای را که می بایست در نظر گرفت این است که داده هایی که با آن مدل اصلی آموزش داده شده به داده های ما تقریبا شبیه باشد چرا که هدف اصلی استخراج نکردن مجدد فیچرهای پایه ای است که این فیچرها در لایه های پایین شبکه تولید میشوند .

در حل این مسئله از مدل از پیش تنظیم شده Inception استفاده شده است.

## 4-9- مدل Inception

ابتدا معماری شبکه های عصبی کانولوشنی (CNN) معمولی را مرور کرده و مصالحه‌ای که در ساخت همه‌ی شبکه‌ها باید برقرار شود را مورد مطالعه قرار دهیم.

اجزای تشکیل‌دهنده‌ی CNNها عبارت‌اند از:

- لایه‌ی کانولوشن (+ تبدیلات غیرخطی که از طریق توابع فعالسازی اجرا می‌شوند)
- لایه‌ی پولینگ
- لایه‌ی تراکم (کاملاً متصل)
- هر بار بخواهیم یک لایه‌ی جدید قبل از لایه‌های تراکم (که در انتهای شبکه قرار دارند) اضافه کنیم، دو نکته‌ی مهم را باید تعیین کنیم:

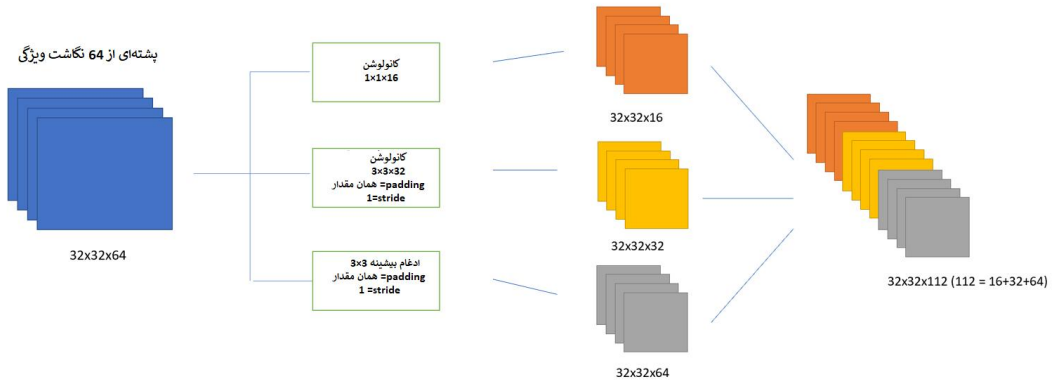
انتخاب بین عملیات کانولوشن و یا ادغام؛

تعیین اندازه و تعداد فیلترهایی که از خروجی لایه‌ی قبلی وارد لایه‌ی جدید خواهند شد.

راهکار ایده‌آل این است که بتوان همه‌ی گزینه‌های موجود را در یک لایه به صورت یک‌جا امتحان کرد. در همین راستا، تیم پژوهشی گوگل، معماری جدیدی طراحی کردند که یک لایه‌ی جدید به نام Inception دارد.

هدف اصلی از طراحی ماژول Inception این بود که چندین عملیات (ادغام، کانولوشن) با فیلترهایی به اندازه‌های گوناگون (۳×۳، ۵×۵ و ...) را بتوان به صورت موازی ایجاد کرد و نیازی به انتخاب بین آن‌ها نباشد.

نحوه‌ی کارکرد ماژول Inception را با هم بررسی می‌کنیم:




شکل (4-7) معماری مدل Inception

همانطور که مشاهده می‌کنید، ورودی اولیه (پشته‌ای از نقشه‌های ویژگی که خروجی لایه‌ی قبلی هستند) تنسور با ۶۴ نقشه‌ی ویژگی است، ابعاد همه‌ی این نگاشت‌ها ۳۲×۳۲ می‌باشد. سه عملیات، به صورت موازی، روی این تنسور اجرا می‌شوند:

- عملیات کانولوشن با ۱۶ فیلتر ۱×۱: اندازه‌ی تنسور خروجی ۱۶×۳۲×۳۲ خواهد بود (عدد آخر، یعنی ۱۶، نشان‌دهنده‌ی تعداد نهایی نقشه‌های ویژگی است که برابر با تعداد فیلترهای اعمال شده روی تصویر می‌باشد).
- عملیات کانولوشن با ۳۲ فیلتر ۳×۳: هدف از این عملیات این است که ابعاد خروجی هم‌اندازه با نگاشت‌های ویژگی اصلی باقی بماند. padding را می‌توان برابر با ۱ و stride (گام) را برابر با ۱ قرار داد (برای کسب اطلاعات بیشتر در مورد padding و strides و تأثیرات آن‌ها روی ابعاد نگاشت‌ها به این مقاله مراجعه کنید). اندازه‌ی تنسور خروجی ۳۲×۳۲×۳۲ خواهد بود.
- عملیات پولینگ ماکزیمم با یک فیلتر ۳×۳ (مقادیر padding و stride طبق استدلال بیان شده در عملیات

قبلی محاسبه می‌شوند): اندازه‌ی تنسور خروجی ۶۴×۳۲×۳۲ خواهد بود؛ از آن‌جایی که فیلتر پولینگ روی همه‌ی نقشه‌های ویژگی تنسور ورودی اجرا می‌شود، عمق تنسور خروجی برابر با عمق تنسور اصلی (=۶۴) است.

بدیهی است که با افزودن این عملیات‌ها به تمام لایه‌ها، مدل از نظر تعداد پارامترها پیچیده‌تر می‌شود. اما خوشبختانه، نسخه‌ی دوم ماژول Inception تکنیک خوبی برای کاهش ابعاد نگاشت‌های ویژگی قبل از اجرای مدل ارائه داده است.

# فصل 5:

# پیاده سازی روش پیشنهادی

## 5-1- داده های متنی

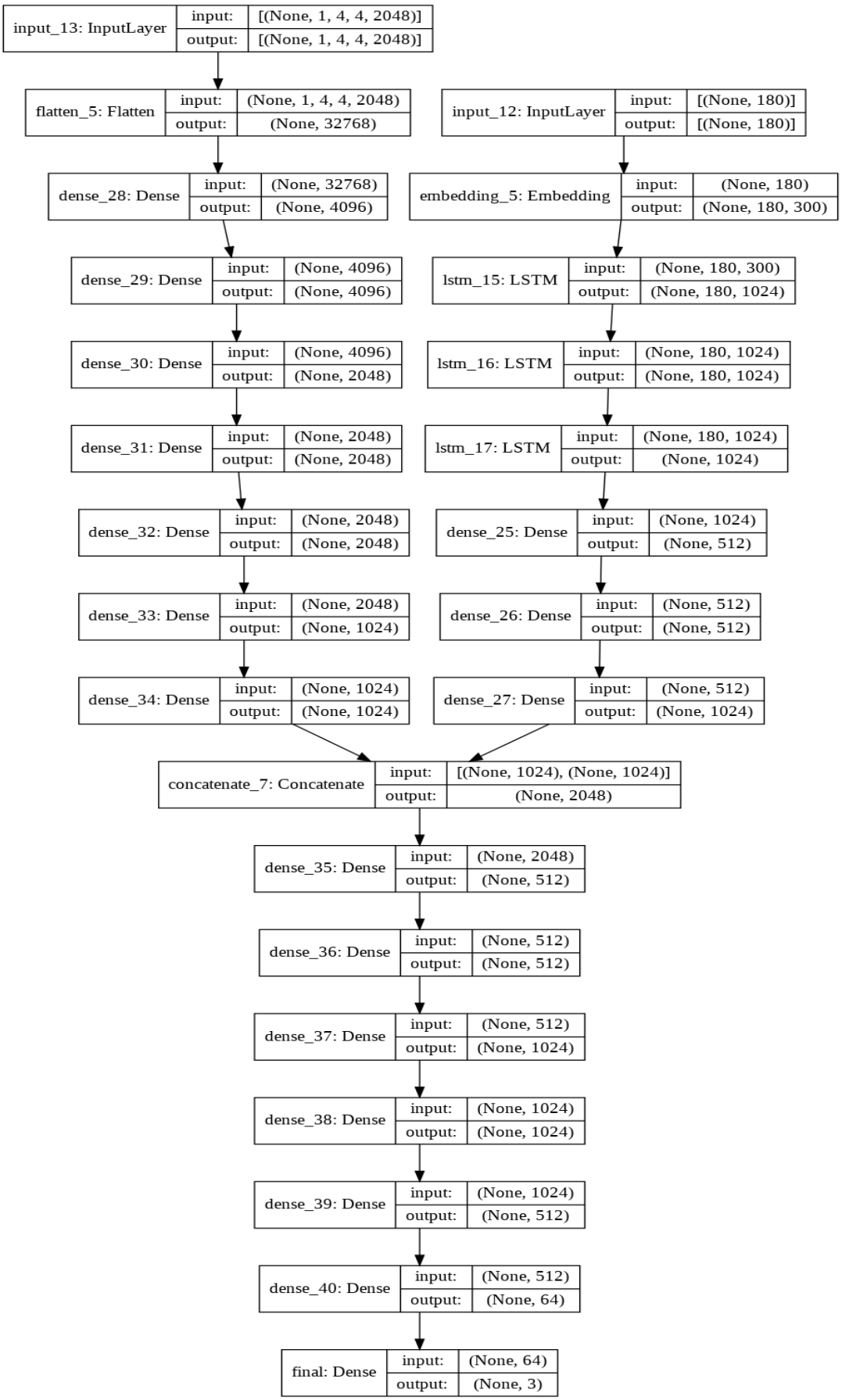
برای پیاده سازی مدل مورد نظر ابتدا داده های متنی استخراج شده به فضای embedding درآمده اند و سپس به عنوان ورودی به شبکه های به هم پیوسته LSTM با خروجی نهایی ۱۰۲۴ نورون داده شده اند. برای embedding ابتدا کلمه ها توکنایز شده اند و سپس به کمک کتابخانه glove به فضای ۳۰۰ بعدی درآمده اند.

## 5-2- داده های تصویری

داده های تصویری از گوگل درایو خوانده شده و سایز آنها به سایز ورودی مدل Inception که به صورت (1,4,4,2048) میباشد تغییر یافته است. پس از عبور داده های ورودی از مدل Inception خروجی به صورت ۴۰۹۶ نورون در آمده است. فیچر های استخراج شده به عنوان ورودی به شبکه پرسپترون چند لایه وارد شده و خروجی ۱-۲۴ نورون ساخته میشود.

## 5-3- ادغام خروجی ها

خروجی دو شاخه مذکور با هم ادغام شده و به عنوان ورودی وارد شبکه پرسپترون دیگری شده که خروجی نهایی ۳ نورون (که بیانگر مفهوم منفی و مثبت و خنثی میباشد) را تشکیل میدهد.




شکل (5-1) شمای کلی مدل پیشنهادی

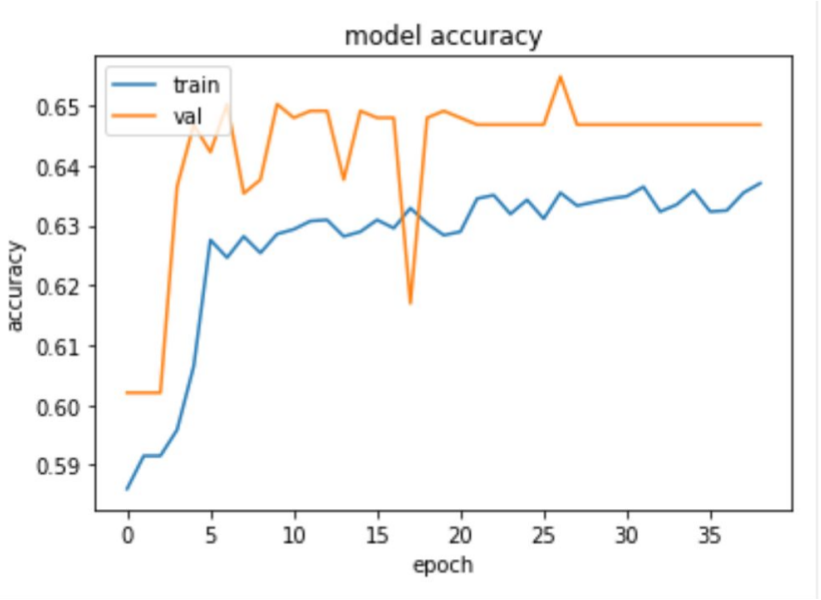
## 5-4- یادگیری

- در فرایند یادگیری تکنیک ها و نکاتی استفاده شده اند که به ما کمک میکند دچار overfitting نشویم:
  استفاده از زمانبند (Scheduler) استفاده شده که نرخ تعلیم را هر ۳۰ ایپاک ۱/۱۰ کاهش میدهد.
- همچنین از توقف زودهنگام (early stopping) نیز استفاده شده است که در هنگام مشاهده کاهش دقت فرایند یادگیری را متوقف میکند.
- به دلیل بالانس نبودن وزن داده ها (از نظر خنثی مثبت و منفی بودن) به کلاس ها وزن داده شده است.

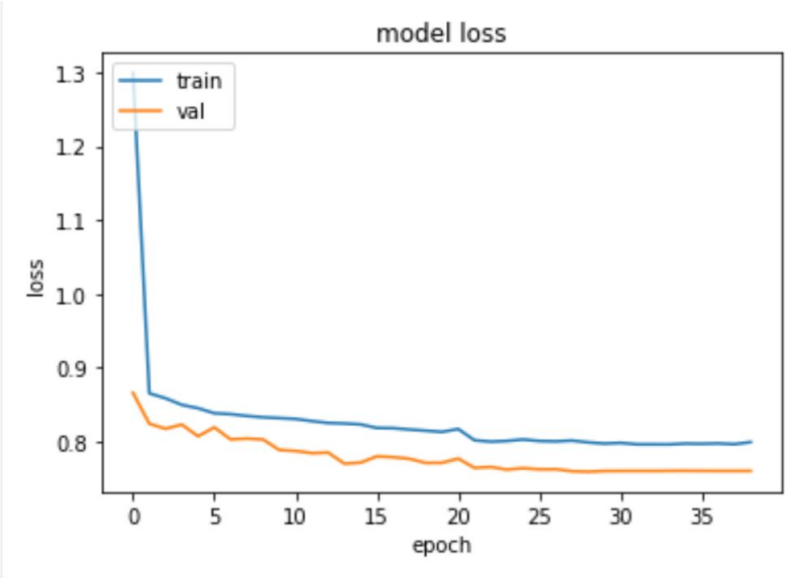
## 5-5- ارزیابی

بین تمامی داده های موجود ۲۹۷۱ عکس متعلق به دسته ی مثبت، ۱۷۲۰ عکس خنثی و ۳۳۶ عکس منفی میباشند. اگر مدل به عنوان روش پایه تمامی عکس های ورودی را مثبت دستهبندی کند، دقت پایه ۵۹.۱۰٪ خواهد بود.

پس از آموزش مدل روی داده های تمرینی، دقت که به آن دست یافته شده ۶۶٪ میباشد. این بدین معنی است که مدل از مدل پایه بهتر عمل کرده است و حتما جای پیشرفت دارد.




شکل (5-2) دقت مدل حین یادگیری (با داده ولیدیشن)




شکل (5-2) خطای مدل حین یادگیری (با داده ولیدیشن)

# فصل 6:

# نتیجه گیری و کارهای آینده

## 6-1- نتیجه گیری

در این مقاله مدلی به کمک تکنیک های یادگیری عمیق جهت تشخیص مفهوم کلی عکس پیاده سازی کردیم.

مدل ذکر شده از ادغام دو representation مختلف از فیچر های متن و خود عکس ساخته شده است. از چالش های پیش رو کلمات و جمله های نویزی و بدون معنا و سختی ذاتی تشخیص طنز و ... را میتوان نام برد.

## 6-2- کارهای آینده

از کار هایی که برای پیشرفت در زمینه تشخیص تصاویر خنده دار اینترنتی میتوان نام برد:

- ساخت و جمع آوری دیتاست بهتر و به زبان های مختلف
- پیشنهاد دادن تصاویر به کاربران فضای مجازی به عنوان پروژه مرتبط
- بکار بردن مدل های از پیش تعلیم داده شده دیگری مانند BERT و ...

# مراجع

[1] Nikhil Sonnad. The world's biggest meme is the word "meme" itself. 2018.

[2] Noam Gal, Limor Shifman, and Zohar Kampf. "it gets better": Internet memes and the construction of collective identity. New Media & Society, 18(8):1698–1714, 2016.

[3] Amanda Williams, Clio Oliver, Katherine Aumer, and Chanel Meyers. Racial microaggressions and perceptions of internet memes. Computers in Human Behavior, 63:424–432, 2016. 12 A PREPRINT - AUGUST 11, 2020

[4] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. SemEval2019 Task 6: Identifying and categorizing offensive language in social media (OffensEval). In 13th International Workshop on Semantic Evaluation, pages 75–86. ACL, 2019.

[5] V Peirson, L Abel, and E Meltem Tolunay. Dank learning: Generating memes using deep neural networks. arXiv preprint arXiv:1806.04510, 2018.

[6] Hugo Gonc¸alo Oliveira, Diogo Costa, and Alexandre Miguel Pinto. One does not simply produce funny memes!– explorations on the automatic generation of internet humor. In 7th International Conference on Computational Creativity, pages 238–245, Paris, France, 2016. Sony CSL.

[7] Jean H French. Image-based memes as sentiment predictors. In 2017 International Conference on Information Society (i-Society), pages 80–85. IEEE, 2017.

[8] Viswanath Sivakumar, Albert Gordo, and Manohar Paluri. Rosetta: Understanding text in images and videos with machine learning. 2018.

[9] Sicheng Zhao, Guiguang Ding, Tat-Seng Chua, Bjorn Schuller, and Kurt Keutzer. Affective image content ¨ analysis: A comprehensive survey. pages 5534–5541, 07 2018.

[10] Jeffrey Pennington, Richard Socher, and Christoper Manning. Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01 2014.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for

large-scale image recognition. CoRR, abs/1409.1556, 2015.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[14] Wasifur Rahman, Md Kamrul Hasan, Amir Zadeh, Louis-Philippe Morency, and Mohammed Ehsan Hoque. M-bert: Injecting multimodal information in the bert structure, 2019.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[16] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[17] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

[18] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. CoRR, abs/1602.07261, 2016.

[19] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks, 2016.

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. CoRR, abs/1709.01507, 2017.

[21] Chenxi Liu, Barret Zoph, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan L. Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. CoRR, abs/1712.00559, 2017.

[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[23] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.

[24] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. CoRR, abs/1906.08237, 2019.

[25] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227, 2014.

[26] Jorge Ramon Fonseca Cacho, Kazem Taghva, and Daniel Alvarez. Using the google web 1t 5-gram corpus ´ for ocr error correction. In 16th International Conference on Information Technology-New Generations (ITNG 2019), pages 505–511. Springer, 2019. 13 A PREPRINT - AUGUST 11, 2020

[27] Jamshed Memon, Maira Sami, and Rizwan Ahmed Khan. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). arXiv preprint arXiv:2001.00139, 2020.

[28] Noman Islam, Zeeshan Islam, and Nazia Noor. A survey on optical character recognition system. arXiv preprint arXiv:1710.05703, 2017.

[29] Joanna Isabelle Olszewska. Active contour based optical character recognition for automated scene understanding. Neurocomputing, 161:65–71, 2015.

[30] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes, 2020.