





دانشگاه سمنان

دانشگاه سمنان

دانشکده برق و کامپیوتر

گزارش تمرین آموزش دادن روش resnet50 بر روی مدل FGVC-Aircraft

نگارش

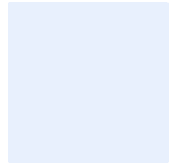
علی مولائی

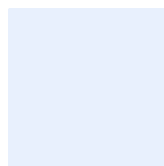
استاد راهنما

دکتر محمد جواد فدائی اسلام

استاد مشاور

اردیبهشت 1402





چکیده

هدف: پیاده سازی مدل **resnet50** بر روی مدل آماده FGVC-Aircraft
روش شناسی پژوهش: تنها تکنیک ها یا رویکردهایی که به ضرورت، برای درک پژوهش الزامی است
در این جا تشریح می شود.

یافته ها:

نتیجه گیری: با استفاده از یادگیری انتقالی میتوان داده هایی که سنگین نیستند را آموزش داد و پردازش
کرد.

کلیدواژه ها: resnet,transflow,googlenet

فهرست نوشتار

10.....	1.مقدمه.....
17.....	2.. یادگیری انتقالی.....
26.....	2-1. چرا یادگیری انتقالی.....
29.....	2-1. انواع یادگیری انتقالی.....
36.....	3. یادگیری انتقالی resnet.....
40.....	2-1. یادگیری انتقالی resnet.....
42.....	4. پیاده سازی آموزش FGVC-Aircraft با استفاده از resnet50.....
48.....	5. فهرست منابع.....

1. مقدمه

قبل از اینکه وارد یادگیری انتقالی بشیم بگذارید درباره یادگیری ماشین، یادگیری عمیق، شبکه عصبی و بینایی ماشین پردازیم تا مطالب را بهتر متوجه بشید.

1- یادگیری ماشین:

یادگیری ماشین (Machine Learning) یا به اختصار ML به سیستم‌ها کمک می‌کند تا به صورت خودکار یادگیری و پیشرفت داشته باشند. یک ماشین به کمک ماشین لرنینگ می‌تواند از تجربیات و مشاهداتی که بر اساس یک مجموعه داده تجزیه و تحلیل می‌کند، آموزش ببیند. در این مطلب توضیح می‌دهیم یادگیری ماشین چیست و علت اهمیت، انواع و نحوه انتخاب ماشین لرنینگ مناسب را بررسی می‌کنیم. همچنین بررسی می‌کنیم که چه کسانی از یادگیری ماشین استفاده می‌کنند و آینده‌ی آن چگونه خواهد بود.

الگوریتم‌های یادگیری ماشین از داده‌های سابق به عنوان ورودی برای یادگیری و سپس پیش بینی مقادیر خروجی جدید استفاده می‌کنند.

2- یادگیری عمیق

یادگیری عمیق به عنوان یکی از زیرشاخه‌های حوزه یادگیری ماشین تلقی می‌شود. هدف یادگیری عمیق طراحی سیستم‌های کامپیوتری هوشمندی است که بتوانند مشابه انسان درباره موضوعی خاص، راه حل ارائه کنند و مفاهیم جدیدی را یاد بگیرند.

این حوزه از فناوری، شاخه‌ای مهم در «علم داده» (Data Science) است، زیرا اصلی‌ترین مباحث این شاخه، آمار و مدل‌سازی برای پیش‌بینی مسائل مختلف را شامل می‌شود. مهندسان علم داده با استفاده از روش‌های یادگیری عمیق می‌توانند جمع‌آوری، تجزیه و تحلیل و تفسیر حجم عظیمی از داده‌ها را سریع‌تر و آسان‌تر انجام دهند.

الگوریتم‌های یادگیری عمیق با در اختیار داشتن ورودی‌های مختلفی از دنیای بیرون مانند تصاویر، صوت و متن، به دنبال پیدا کردن الگوهایی هستند که با استفاده از آن‌ها بتوانند پیش‌بینی خاصی را پیرامون موضوع مطرح شده انجام دهند.

به منظور ارائه پاسخ دقیق به پرسش یادگیری عمیق چیست باید نگاهی به ساختار مدل‌های این حوزه از فناوری پرداخت. مدل‌های یادگیری عمیق از ساختارهای لایه به لایه‌ای تشکیل شده‌اند که به آن‌ها «شبکه عصبی» (Neural Network) گفته می‌شود. طرح چنین ساختاری، الهام گرفته از مغز انسان است. «نورون‌های» (Neurons) مغز انسان دارای اجزایی با نام‌های «دندریت» (Dendrite)، «هسته» (Nucleus)، «جسم سلولی» (Cell Body)، «آکسون» (Axon) و «پایانه‌های آکسون | سیناپس» (Axon Terminals | Synapse) هستند. ورودی‌های نورون‌ها، سیگنال‌هایی به حساب می‌آیند که از حس‌های بینایی، شنوایی، بویایی و لامسه انسان دریافت شده و به دندریت منتقل می‌شوند و سپس این اطلاعات به سمت آکسون ارسال شده و در نهایت از طریق پایانه‌های سیناپس به دندریت نورون بعدی فرستاده می‌شوند.

شبکه‌های عصبی نیز همانند مغز انسان، ورودی‌ها را که در قالب بردارهای عددی هستند، در هر لایه با استفاده از چندین «گره» (Nodes) دریافت می‌کنند و با اعمال پردازش بر روی آن‌ها، خروجی حاصل شده را به لایه بعدی خود منتقل می‌کنند. وزن‌های موجود در شبکه عصبی، نقش سیناپس را در مغز انسان ایفا می‌کنند. به عبارتی، وزن‌ها همان چیزی هستند که شبکه‌های

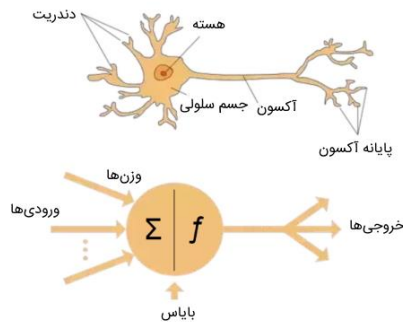
عصبی باید یاد بگیرند. شبکه عصبی با استفاده از وزن‌های یاد گرفته شده، تصمیم می‌گیرد کدام یک از ورودی‌های نورون‌ها در تعیین خروجی شبکه از اهمیت بالایی برخوردار هستند.

داده‌ها پس از گذشتن از تمامی لایه‌های میانی شبکه عصبی، به لایه آخر منتقل می‌شوند تا خروجی نهایی شبکه توسط این لایه محاسبه شود. خروجی‌های شبکه می‌توانند در قالب‌های مختلفی ارائه شوند که هر یک از آن‌ها در ادامه فهرست شده‌اند:

خروجی شبکه عصبی می‌تواند از نوع اعداد «پیوسته» (Continuous) باشند. قیمت‌های محصولات می‌تواند از نوع اعداد پیوسته محسوب شوند.

شبکه عصبی می‌تواند خروجی را در قالب اعداد دودویی و به شکل 0 و 1 ارائه دهد.

خروجی شبکه عصبی را می‌توان به صورت چندین دسته مشخص کرد. به عنوان مثال، برای پژوهشی پیرامون تشخیص انواع تصاویر حیوانات، می‌توان چندین دسته مشخص کرد که هر یک از دسته‌ها، نام حیوان خاصی مانند موش، گربه، سگ، فیل، مار و اسب را مشخص می‌کند.



شباهت نورون در مغز و گره در شبکه عصبی

3- بینایی ماشین:

بینایی ماشین یک فناوری پیشرفته و یکی از شاخه‌های رو به رشد هوش مصنوعی به‌شمار می‌آید. این تکنولوژی مبتنی بر پردازش تصاویر دو بعدی است و به کامپیوتر قابلیت مشاهده و تجزیه تحلیل محیط پیرامون را می‌دهد. به عبارتی ساده‌تر بینایی ماشین یک قابلیت کامپیوتری مجهز به چندین دوربین فوق پیشرفته و یک سیستم تحلیلگر قدرتمند برای پردازش داده‌های تصویری است. بینایی ماشین یکی از مباحث علوم میان رشته‌ای محسوب می‌شود که با علوم مختلفی مانند صنایع، مکانیک، برق و الکترونیک و علوم رایانه، همپوشانی دارد در واقع بینایی ماشین شاخه‌ای از هوش مصنوعی است که از مبانی علوم رایانه، تکنیک‌های پردازش تصویر، مهندسی اپتیک، مهندسی مکانیک استفاده می‌کند و با اتوماسیون صنعتی ارتباط تنگاتنگی دارد.

امروزه سیستم‌های بینایی ماشین از طریق اصلاح و بهینه‌سازی تکنیک‌های موجود در حوزه‌های مرتبط به سرعت در حال پیشرفت هستند و در نتیجه زیرشاخه‌های تحقیقاتی بسیاری در حوزه بینایی ماشین در حال شکل‌گیری هستند که محققان با پرداختن به آنها می‌توانند عملکرد بینایی ماشین را بیشتر از گذشته بهبود بخشند. در مقاله پیش رو سعی بر این است که علاوه بر معرفی بینایی ماشین و تفاوت‌های آن با مفاهیم کاربردی دیگر در حوزه هوش مصنوعی به بیان کاربردهای آن در زمینه‌های مختلف صنعتی و علمی پردازیم.

“بینایی ماشین” یا “Machine Vision” یک قابلیت کامپیوتر برای مشاهده محیط پیرامون است که عمدتاً از یک یا چند دوربین ویدئویی با قابلیت تبدیل آنالوگ به دیجیتال و همچنین یک سیستم تحلیل سیگنال دیجیتال استفاده می‌کند. در این حالت داده‌های تصویری یا همان Image Data به کامپیوتر یا کنترلر سیستم ارسال می‌شود.

3. یادگیری انتقالی

تکنیک یادگیری انتقالی از دانش مدلی که از قبل برای وظیفه‌ای دیگر (تسک اولیه) آموزش دیده شده است برای حل وظیفه‌ای دیگر استفاده می‌کند. قطعاً این دو وظیفه تا حد زیادی مشابه یکدیگر هستند؛ برای مثال، اگر یک طبقه‌بندی‌کننده ساده را برای پیش‌بینی اینکه در عکس کوله‌پشتی وجود دارد یا نه آموزش داده‌ایم، می‌توانیم از دانشی که مدل در طول آموزش به دست آورده است برای پیش‌بینی وجود اشیای دیگر، مانند عینک آفتابی در عکس، استفاده کنیم. درواقع با تکنیک یادگیری انتقالی ما وزن‌هایی که یک شبکه برای وظیفه‌ی A یاد گرفته است به وظیفه‌ی B منتقل می‌کنیم.



طریقه کار یادگیری انتقالی

3-1. چرا یادگیری انتقالی؟

هم‌ترین مزایای استفاده از تکنیک یادگیری انتقالی (Transfer learning) این است که زمان آموزش به شدت کاهش پیدا می‌کند، به حجم زیادی داده احتیاج نداریم و عملکرد شبکه‌ی عصبی در اغلب موارد بهتر خواهد بود.

برای آموزش شبکه‌های عصبی به حجم زیادی داده نیاز داریم که این حجم از داده همیشه

در دسترس نیست. اینجاست که یادگیری انتقالی می تواند کمک بزرگی باشد. آموزش شبکه از ابتدا به زمان زیادی نیاز دارد و گاهی ممکن است در وظیفه های پیچیده این موضوع هفته ها طول بکشد. استفاده از یادگیری انتقالی این زمان را کاهش می دهد؛ زیرا ما از شبکه ای که از قبل آموزش دیده است استفاده می کنیم.

3-2. انواع یادگیری انتقالی

آموزش یک مدل برای استفاده ی مجدد

فرض کنیم که ما برای انجام دادن وظیفه ی A به حجم زیادی داده نیاز داریم، اما این حجم از داده در دسترس نیست. در این حالت می توانیم مدل را برای وظیفه ی مشابه دیگری (B) که داده های در دسترس زیادی دارد آموزش دهیم؛ سپس از این مدل آموزش دیده برای تسک مدنظر خودمان (A) استفاده کنیم. اینکه از کل مدل یا فقط چندین لایه ی آن استفاده کنیم به وظیفه ای بستگی دارد که می خواهیم انجام دهیم.

استفاده از مدل های ازپیش آموزش دیده (Pre-trained Models)

یکی دیگر از روش های تکنیک یادگیری انتقالی استفاده از مدل های ازپیش آموزش دیده ی موجود است. تعداد زیادی از این مدل ها در حال حاضر در دسترس هستند و با کمی تحقیق می توان از آن ها برای وظیفه ی مدنظر خود استفاده کنیم. این نوع از یادگیری انتقالی بیشترین کاربرد را در یادگیری عمیق دارد.

استخراج ویژگی (Feature Extraction)

یکی دیگر از روش های یادگیری انتقالی استخراج مهم ترین ویژگی ها با استفاده از یادگیری عمیق است. در یادگیری ماشین معمولاً متخصصان ویژگی ها را به صورت دستی ایجاد می کنند، اما خود شبکه ی عصبی هم این قابلیت را دارد که تشخیص دهد کدام ویژگی ها مهم هستند و کدام یک اهمیت ی ندارد. شبکه ی عصبی می تواند در کم ترین (بهینه ترین) زمان ممکن ترکیب خوبی از ویژگی ها را حتی برای پیچیده ترین وظیفه ها در اختیارمان قرار دهد. این کار به صورت

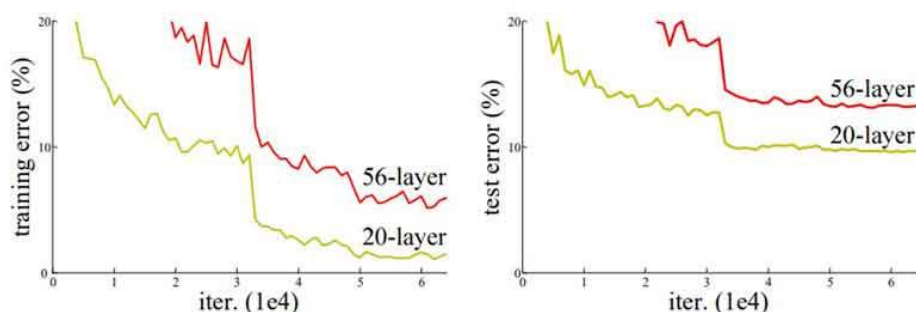
دستی زمان و زحمت زیادی را می‌طلبد. این ویژگی‌های یادگرفته‌شده می‌توانند برای تسک‌های دیگری هم استفاده شوند. می‌توانیم از لایه‌های اولیه‌ی شبکه برای شناسایی ویژگی‌ها استفاده کنیم و لایه‌های نهایی را مطابق با وظیفه‌ی خود تغییر دهیم. خروجی این شبکه بردارهای ویژگی داده‌های خام ورودی را به ما می‌دهد.

3. یادگیری انتقالی Resnet

3-1. یادگیری انتقالی Resnet چیست؟

شبکه Resnet که مخفف Residual Network (شبکه باقی مانده) است نوع خاصی از شبکه عصبی است که در سال 2015 توسط Kaiming He, Xiangyu Zhang و Shaoqing Ren و Jian Sun در مقاله‌ای با عنوان «یادگیری باقیمانده عمیق برای تشخیص تصویر» معرفی شد. مدل‌های Resnet به موفقیت‌های چشمگیری دست یافتند. از جمله‌ی این موفقیت‌ها، کسب مقام اول در رقابت ILSVRC 2015 Classification بود. نوآوری اصلی این شبکه آن بود که برای حل مشکل زیاد شدن تعداد لایه‌ها، بلوک جدیدی به نام بلوک باقیمانده (Residual Block) معرفی شد که در ادامه توضیح داده خواهد شد.

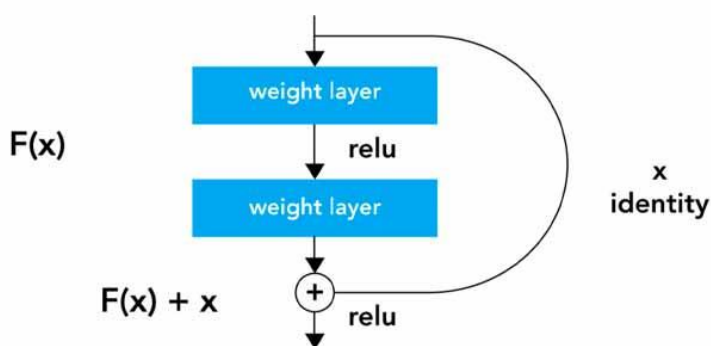
در اکثر مواقع برای حل یک مسئله پیچیده، چند لایه به معماری شبکه عصبی خود اضافه می‌کنیم و به اصطلاح شبکه خود را عمیق‌تر می‌کنیم که در نهایت دقت و عملکرد شبکه را افزایش می‌دهد. منطق اصلی برای اضافه کردن لایه‌های بیشتر آن است که با اضافه شدن این لایه‌ها، شبکه به تدریج ویژگی‌های پیچیده‌تری را یاد می‌گیرد. برای مثال در مسئله‌ی تشخیص تصاویر، لایه‌ی اول ممکن است لبه‌های تصویر را تشخیص دهد، لایه‌ی دوم زمینه را یاد بگیرد و به طور مشابه لایه‌ی سوم می‌تواند کل ساختار شی یا چهره را یاد بگیرد. محققان با انجام آزمایش‌های متعدد متوجه شده‌اند که در مدل شبکه عصبی کانولوشنی، آستانه‌ای برای افزایش عمق وجود دارد. در ادامه نموداری آمده است که نشان می‌دهد درصد خطا برای داده‌های آموزش و آزمایش به ترتیب برای شبکه‌های 20 لایه و 56 لایه چقدر است.



همانطور که در شکل فوق مشخص است، درصد خطا برای هر دو داده‌ی آزمایش و آموزش برای یک معماری 56 لایه بیشتر از یک شبکه 20 لایه است. این نشان می‌دهد که اضافه کردن لایه‌های بیشتر در شبکه همیشه به معنای بهتر شدن عملکرد آن نیست بلکه ممکن است عملکرد آن را تخریب کند. علت اینکه با عمیق‌تر شدن شبکه، عملکرد آن ضعیف می‌شود را می‌توان در عوامل متعددی جستجو کرد مانند تابع بهینه سازی، وزن دهی اولیه شبکه و مهم‌تر از همه مسئله ی *gradient vanishing*.

بلوک Residual

مشکلات مربوط به آموزش شبکه‌های بسیار عمیق با معرفی شبکه‌های ResNet تا حدودی حل شد. این معماری Resnet از بلوک‌های Residual تشکیل می‌شد که در شکل زیر مشاهده می‌کنید.



با مشاهده شکل فوق، اولین تفاوتی که متوجه می‌شویم آن است که یک اتصال مستقیم وجود دارد که لایه‌ی میانی (در معماری بزرگتر، لایه‌های میانی) را رد کرده است. این اتصال "*Skip Connection*" نام دارد و عنصر اصلی بلوک‌های باقیمانده است. به دلیل وجود این اتصال، خروجی لایه مشابه شبکه‌های دیگر نیست. ورودی " x " در وزن لایه‌ها ضرب شده و همچنین مقدار بایاس به آن اضافه می‌شود. سپس این عبارت از تابع فعالسازی $F(x)$ عبور می‌کند و در نهایت خروجی را داریم که با $H(x)$ نمایش می‌دهیم. و به صورت زیر تعریف می‌شود.

$$H(x)=f(x)+x$$

در شبکه‌های عصبی عادی، لایه‌ها توزیع ورودی را یاد می‌گیرند اما در این بلوک‌های *Residual* شبکه توزیع خروجی-ورودی را یاد می‌گیرد و به همین دلیل به این بلوک‌ها، بلوک‌های باقیمانده گفته می‌شود.

شبکه‌ی ResNet چطور مشکلات ناشی از عمیق شدن شبکه را حل می‌کند؟

همانطور که گفته شد در شبکه‌ی ResNet، اتصالی به نام Skip Connection وجود دارد که مسئله‌ی Vanishing gradient را در شبکه‌های عصبی عمیق حل می‌کند و امکانی را فراهم می‌آورد که گرادینان از مسیرهای Shortcut عبور کنند. همچنین با وجود این اتصالات، مدل می‌تواند تابع همانی را یاد بگیرد که تضمین می‌کند لایه‌ی بالاتر حداقل به خوبی لایه‌ی پایین‌تر عمل می‌کنند، نه بدتر از آن.

فرض کنید ما یک شبکه‌ی کم عمق و یک شبکه‌ی عمیق داریم که ورودی " x " را به خروجی " y " با استفاده از تابع $H(x)$ نگاشت می‌کنند. هدف ما آن است که شبکه عمیق حداقل به خوبی شبکه کم عمق کار کند و عملکرد شبکه با زیاد شدن تعداد لایه‌ها، پایین نیاید. یک راه برای دستیابی به این هدف آن است که لایه‌های اضافی در شبکه عمیق، تابع همانی را یاد بگیرند و بنابراین خروجی آنها برابر با ورودی آنها باشد و در نهایت با وجود لایه‌های اضافی، عملکرد مدل تخریب نشود.

برای آنکه شبکه، تابع همانی را یاد بگیرد، باید $F(x)=0$ باشد آنگاه رابطه به صورت زیر خواهد شد

$$H(x)=x$$

در بهترین حالت، لایه‌های اضافی شبکه عصبی عمیق می‌توانند نگاشت ورودی " x " به خروجی " y " را نسبت به شبکه‌های کم عمق خود بهتر تخمین بزنند و خطا را با حاشیه‌ی قابل توجه و معناداری کاهش دهند. بنابراین انتظار داریم که شبکه ResNet به خوبی یا بهتر از شبکه‌های عصبی عمیق کار کند

فصل چهارم. پیاده سازی تشخیص هواپیماها با استفاده از resnet50

معماری resnet50 :

معماری ResNet-50 را می توان به 6 قسمت تقسیم کرد

پیش پردازش ورودی

1-بلوک های Cfg[0]

2-بلوک های Cfg[1]

3-بلوک های Cfg[2]

4-بلوک های Cfg[3]

لایه کاملاً متصل

نسخه های مختلف معماری ResNet از تعداد متفاوتی از بلوک های Cfg در سطوح مختلف استفاده می کنند، همانطور که در شکل بالا ذکر شد. فهرست دقیق و آموزنده را می توان در زیر یافت.

برای پیاده سازی معماری ResNet باید از پکیج tensorflow در پایتون استفاده کرد

که ما با مراجعه به منابع مختلف نرم افزار تشخیص مدل های هواپیما را پیاده سازی کرده و نتایج را این گونه مثل جدول زیر میتوان مشاهده کرد.

id	families	address
0034958	A300	fgvc-aircraft-2013b/data
0037511	Boeing 767	fgvc-aircraft-2013b/data

چون برنامه در محیط کنسول پیاده شده امکان دیدن فیزیکی تصاویر وجود ندارد فقط نتایج از طریق جدول مشخص میشود اما دیتاها دیتاهای داخلی است مثلاً متوانید طبق عکس هایی که داریم بفهمیم مدل این هواپیما های زیر چیست.



برخی توابع اصلی این سیستم را در ایت داکيومن قرار مدیدیم و برای دیدن سورس کد به پیوست مراجعه کنید.

کد خواندن دیتا ها از لیست و پوشه عکس ها

```
train = pd.read_csv("images_train.csv")
#get image path
def get_paths(sub):
    index = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "a", "b", "c", "d", "e", "f"]

    paths = []

    for a in index:
        try:
            paths.extend([f'../fgvc-aircraft-2013b/data/images" + x for x in os.listdir(f'../fgvc-airc
        except:
            pass
```

کد شناختن بلوک:

```
def identity_block(X, f, filters, stage, block)
'conv_name_base = 'res' + str(stage) + block + '_branch
'bn_name_base = 'bn' + str(stage) + block + '_branch

F1, F2, F3 = filters

X_shortcut = X

X = Conv2D(filters=F1, kernel_size=(1, 1), strides=(1, 1), padding='valid',
            ,name=conv_name_base + '2a
(kernel_initializer=glorot_uniform(seed=0))(X
```

```

X = BatchNormalization(axis=3, name=bn_name_base + '2a')(X)
X = Activation('relu')(X)

X = Conv2D(filters=F2, kernel_size=(f, f), strides=(1, 1), padding='same',
           ,name=conv_name_base + '2b'
           (kernel_initializer=glorot_uniform(seed=0))(X)
X = BatchNormalization(axis=3, name=bn_name_base + '2b')(X)
X = Activation('relu')(X)

X = Conv2D(filters=F3, kernel_size=(1, 1), strides=(1, 1), padding='valid',
           ,name=conv_name_base + '2c'
           (kernel_initializer=glorot_uniform(seed=0))(X)
X = BatchNormalization(axis=3, name=bn_name_base + '2c')(X)

Add shortcut value to main path #
X = Add()(X_shortcut, X)
X = Activation('relu')(X)

return X

```

کد پیاده سازی 5resnet این پروژه

```

def ResNet50(input_shape = (64, 64, 3), classes = nlabls)
    X_input = Input(input_shape)
    X = ZeroPadding2D((3, 3))(X_input)
    X = Conv2D(64, (7, 7), strides = (2, 2), name = 'conv1', kernel_initializer
              = glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis = 3, name = 'bn_conv1')(X)
    X = Activation('relu')(X)
    X = MaxPooling2D((3, 3), strides=(2, 2))(X)
    X = convolutional_block(X, f = 3, filters = [64, 64, 256], stage = 2,
                          block='a', s = 1)
    X = identity_block(X, 3, [64, 64, 256], stage=2, block='b')
    X = identity_block(X, 3, [64, 64, 256], stage=2, block='c')
    X = convolutional_block(X, f = 3, filters = [128, 128, 512], stage = 3,

```

```

                                block='a', s = 2)
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='b')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='c')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='d')
    X = convolutional_block(X, f = 3, filters = [256, 256, 1024], stage = 4,
                                block='a', s = 2)
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='b')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='c')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='d')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='e')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='f')
    X = convolutional_block(X, f = 3, filters = [512, 512, 2048], stage = 5,
                                block='a', s = 2)
    X = identity_block(X, 3, [512, 512, 2048], stage=5, block='b')
    X = identity_block(X, 3, [512, 512, 2048], stage=5, block='c')
    X = AveragePooling2D(pool_size=(2, 2),name='avg_pool')(X)
                                X = Flatten()(X)
    X = Dense(classes, activation='softmax', name='fc' + str(classes),
                                kernel_initializer = glorot_uniform(seed=0))(X)
    model = Model(inputs = X_input, outputs = X, name='ResNet50')
                                return model

```


5. فصل پنج: عنوان فصل پنج را اینجا وارد کنید

5-1. عنوان فرعی نخست را اینجا وارد کنید.

آنچه در این راهنما می‌آید، دربرگیرنده چگونگی نگارش پاراگراف است. مخاطبان اصلی این راهنما، دانشجویانی هستند که پاراگراف را به زبان فارسی می‌نویسند؛ بنابراین، این راهنما برای زبان فارسی است؛ ولی توصیه‌ها و رهنمودهایی نیز برای دانشجویانی که آنرا به زبان‌هایی همچون عربی، انگلیسی، فرانسه و ... می‌نویسند، در بخش‌های جداگانه آمده است.

آنچه در این راهنما می‌آید، دربرگیرنده چگونگی نگارش پاراگراف است. مخاطبان اصلی این راهنما، دانشجویانی هستند که پاراگراف را به زبان فارسی می‌نویسند؛ بنابراین، این راهنما برای زبان فارسی است؛ ولی توصیه‌ها و رهنمودهایی نیز برای دانشجویانی که آنرا به زبان‌هایی همچون عربی، انگلیسی، فرانسه و ... می‌نویسند، در بخش‌های جداگانه آمده است.

آنچه در این راهنما می‌آید، دربرگیرنده چگونگی نگارش پاراگراف است. مخاطبان اصلی این راهنما، دانشجویانی هستند که پاراگراف را به زبان فارسی می‌نویسند؛ بنابراین، این راهنما برای زبان فارسی است؛ ولی توصیه‌ها و رهنمودهایی نیز برای دانشجویانی که آنرا به زبان‌هایی همچون عربی، انگلیسی، فرانسه و ... می‌نویسند، در بخش‌های جداگانه آمده است.

آنچه در این راهنما می‌آید، دربرگیرنده چگونگی نگارش پاراگراف است. مخاطبان اصلی این راهنما، دانشجویانی هستند که پاراگراف را به زبان فارسی می‌نویسند؛ بنابراین، این راهنما برای زبان فارسی است؛ ولی توصیه‌ها و رهنمودهایی نیز برای دانشجویانی که آنرا به زبان‌هایی همچون عربی، انگلیسی، فرانسه و ... می‌نویسند، در بخش‌های جداگانه آمده است.

آنچه در این راهنما می‌آید، دربرگیرنده چگونگی نگارش پاراگراف است. مخاطبان اصلی این راهنما، دانشجویانی هستند که پاراگراف را به زبان فارسی می‌نویسند؛ بنابراین، این راهنما برای زبان فارسی است؛ ولی توصیه‌ها و رهنمودهایی نیز برای دانشجویانی که آن را به زبان‌هایی همچون عربی، انگلیسی، فرانسه و ... می‌نویسند، در بخش‌های جداگانه آمده است.

6. فهرست منابع

<https://www.kaggle.com/code/sandy1112/create-and-train-resnet50-from-scratch>

<https://cafetadris.com/blog/%DB%8C%D8%A7%D8%AF%DA%AF%DB%8C%D8%B1%DB%8C-%D8%A7%D9%86%D8%AA%D9%82%D8%A7%D9%84%DB%8C-transfer-learning/>

<https://iq.opengenus.org/resnet50-architecture/>

پیوست الف: نمونه یک پیوست

این جانب رضا تهرانی دانش آموخته کارشناسی ارشد رشته زبان و ادبیات فارسی دانشکده ادبیات دانشگاه ایران پدیدآور پایان نامه با عنوان راهنمای نگارش پایان نامه و رساله با راهنمایی دکتر امیر شریف گواهی و تعهد می کنم که بر پایه قوانین و مقررات، از جمله «دستورالعمل نحوه بررسی تخلفات پژوهشی» و همچنین «مصادیق تخلفات پژوهشی» مصوب وزارت علوم، تحقیقات و فناوری (۲۵ اسفند ۱۳۹۳):

- این پایان نامه دستاورد پژوهش این جانب و محتوای آن از درستی و اصالت برخوردار است؛
- حقوق معنوی همه کسانی را که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند، رعایت کرده ام و هنگام کاربرد دستاورد پژوهش های دیگران در آن، با دقت و به درستی به آن ها استناد کرده ام؛
- این پایان نامه و محتوای آن را تاکنون این جانب یا کس دیگری برای دریافت هیچ گونه مدرک یا امتیازی در هیچ جا ارائه نکرده ایم؛
- همه حقوق مادی این پایان نامه از آن دانشگاه ایران است و آثار برگرفته از آن با وابستگی سازمانی دانشگاه ایران منتشر خواهد شد؛

- در همه آثار برگرفته از این پایان نامه، نام استاد(ان) راهنما و اگر استاد راهنمای نخست تشخیص دهد، نام استاد(ان) مشاور و نشانی رایانامه سازمانی آنان را می آورم؛
- در همه گام های انجام این پایان نامه، هرگاه به اطلاعات شخصی افراد یا اطلاعات سازمان ها دسترسی داشته یا آن ها را به کار برده ام، رازداری و اخلاق پژوهش را رعایت کرده ام.

واژه‌نامه توصیفی

پایان‌نامه: واژه پایان‌نامه ریشه‌ای لاتین دارد که به معنای «مسیر» است.

واژه‌نامه فارسی به انگلیسی

پایان‌نامه: Dissertation

رساله: Thesis

واژه‌نامه انگلیسی به فارسی

Dissertation: پایان‌نامه

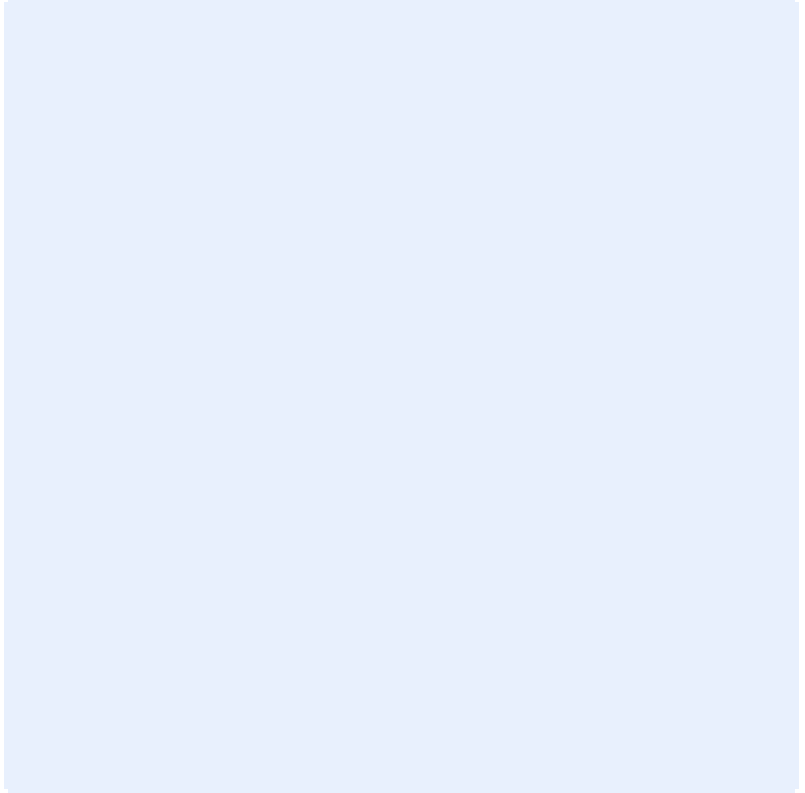
Thesis: رساله

فهرست مقاله‌های برگرفته از پایان‌نامه

رجبعلی بگلو، رضا، رحمت‌الله فتاحی، و مه‌ری پری‌رخ. 1395. «تأثیر نظام‌های اطلاعاتی بر شکل‌گیری مدل‌های ذهنی کاربران نرم‌افزارهای کتابخانه دیجیتال». فصلنامه مطالعات ملی کتابداری و سازماندهی اطلاعات. 27 (2) 21-39.

Rajabali Beglou, Reza, Rahmatollah Fattahi, and Mehri Parirokh. 2018. "Effect of using information systems on the development of mental models among digital library users". *The Electronic Library*, 4 (2), 44-66.

برگ تأیید هیئت داوران / صورت جلسه دفاع (به زبان انگلیسی)



AbstractInsert abstract here.

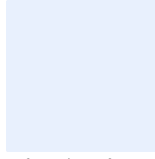
Research Aim: Primary purposes of research, and the research motivation or the reasons why you decided to do this study.

Research method: Describe approaches, methods or techniques whose explains are crucial to understanding the study.

Findings: Findings should be written in a concise and comprehensive manner. This includes experimental and theoretical findings, relations and correlations, observed impacts, etc.

Conclusion: Based on the results of your study, this section should explain the contributions and implications of the research. It may also include suggestions for further studies, evaluations, applications, new relations, and approved & rejected hypothesis.

Keywords:Insert your research keywords here.



Institution/University of ...

Faculty of ...

A Dissertation Submitted in Partial Fulfillment for the Degree of

Insert title of the dissertation.

By

Insert your name and surname here.

Supervisor

Insert the name(s) of supervisor(s) here.

Advisor

Insert the name(s) of advisor(s) here.

Enter month and year of defense (July 2017)