

صلى الله عليه وسلم



Electrical Engineering Department



Amirkabir University of Technology  
(Tehran Polytechnic)

گزارش پروژه درس هوش محاسباتی

استفاده از شبکه رزنت در تشخیص دسته‌بندی تصاویر

## Deep Residual Learning for Image Recognition

نگارندگان:

نیلوفر توحیدی

محمدعرشیا ثمودی

مهبان قلی جعفری

محمد عربزاده

استاد راهنما

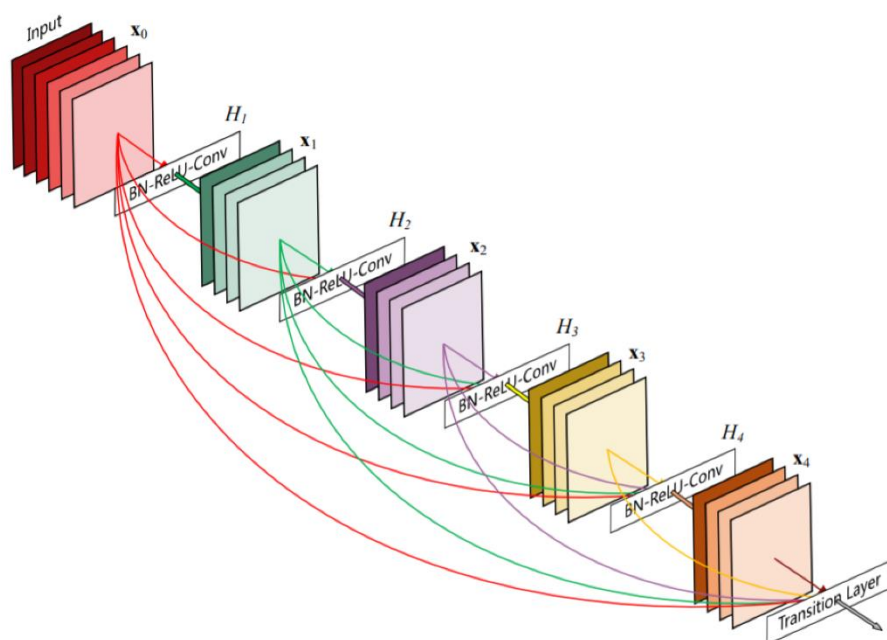
جناب دکتر طالبی

(دانشکده مهندسی برق - دانشگاه امیرکبیر)

پاییز 1400

## چکیده

طی چند سال اخیر، در حوزه‌ی بینایی ماشین (Computer Vision) پیشرفت‌های چشمگیری صورت گرفته است؛ به‌خصوص با معرفی شبکه‌های عصبی عمیق کانولوشن (CNN)، نتایج بسیار خوبی را در زمینه‌ی مسائلی مانند طبقه‌بندی تصاویر (Image Classification) و شناسایی تصاویر (Image Recognition) به دست آورده‌ایم. طی سال‌ها محققان به ایجاد شبکه‌های عصبی عمیق‌تر (افزودن لایه‌های بیشتر) برای حل و بهبود چنین کارهای پیچیده‌ای تمایل پیدا کرده‌اند، اما موضوع این است که با افزودن لایه‌های بیشتری به شبکه‌ی عصبی، آموزش آن‌ها دشوار می‌شود و دقت عملکرد شبکه شروع به کاهش می‌کند. اینجاست که رزنت (ResNet) به کمک‌مان می‌آید و به حل این مشکل کمک می‌کند.



واژه‌های کلیدی: پردازش تصویر ، شبکه‌های عمیق ، رزنت ، Resnet ، طبقه‌بندی تصاویر

## فهرست مطالب

۱	فصل 1: مقدمه
۱-۱	معرفی شبکه‌های عصبی عمیق – Deep Neural Network
۱-۲	معرفی شبکه‌های کانالوشنی - CNN
1-2-1	عملکرد شبکه عصبی کانالوشنی
1-2-2	یک معماری ساده از معماری شبکه عصبی کانولوشن
۱-۳	انواع شبکه‌های کانالوشنی
۱-3-1	LeNet
1-3-2	AlexNet
1-3-3	ZFNET
1-3-4	VGG
1-3-5	GoogLeNet (Inception)
۷	فصل 2: بررسی شبکه Resnet
۲-۱	پیدایش شبکه عصبی باقی‌مانده‌ای
۲-۲	مزیت‌های شبکه Resnet
2-2-1	اتصالات میان‌بر، نقطه‌قوت رزنت
2-2-2	طرز کار شبکه رزنت با اتصالات میان‌بر
2-2-3	معماری شبکه‌ی رزنت (ResNet)
۱۰	فصل 3: پیاده‌سازی شبکه رزنت برای طبقه‌بندی تصاویر دیتاست CIFAR-۱۰
۳-۱	دیتاست CIFAR-۱۰
۳-۲	پیش پردازش داده‌ها
۳-۳	ساختار مدل
۳-۴	آموزش مدل
۱۳	مراجع

## فهرست اشکال

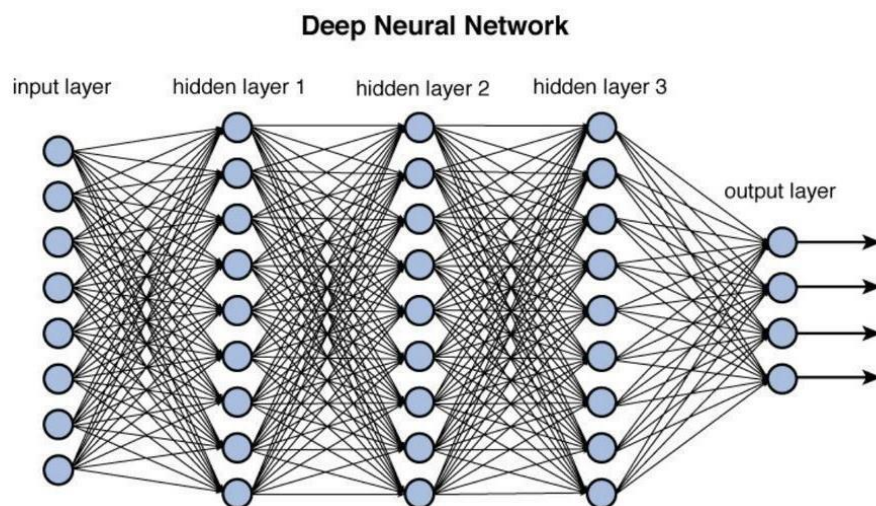
۱	شکل (۱-۱) نمونه‌ای از یک شبکه عمیق.....
۲	شکل (۲-۱) شماتیک نورون مصنوعی.....
۳	شکل (۳-۱) معماری شبکه عصبی کانولوشن.....
۴	شکل (۴-۱) LeNet.....
۴	شکل AlexNet (۱-۵).....
۵	شکل ZFNet (۱-۶).....
۶	شکل (۷-۱) GoogLeNet (Inception).....
۶	شکل (۸-۱) نحوه قرارگیری فیلتر کانالوشنی.....
۸	شکل (۱-۲) اتصال میانبر در زرنت.....
۹	شکل (۲-۲).....
۱۰	شکل (۱-۳) برخی از تصاویر موجود در دیتاست CIFAR-۱۰ به همراه کلاس آن‌ها.....
۱۱	شکل (۲-۳).....
۱۲	شکل (۳-۳).....

# فصل 1: مقدمه

## 1-1- معرفی شبکه‌های عصبی عمیق - Deep Neural Network

در سال‌های گذشته پیشرفت‌های چشم‌گیری در حوزه بینایی ماشین یا Computer Vision اتفاق افتاده است. به خصوص با معرفی شبکه‌های عصبی کانالوشنی یا Convolutional Neural Networks. ما در حال رسیدن به نتایج امیدوار کننده‌ای از مسائل تشخیص طبقه‌بندی عکس‌ها هستیم.

در طی سالیان، محققان با اضافه کردن لایه به شبکه‌های عصبی به مفهومی تحت عنوان شبکه عصبی عمیق یا Deep Neural Network رسیدند که که بتوان با آن‌ها مسائل پیچیده‌تر را حل کرد و بتوان دقت دسته‌بندی‌ها را نیز افزایش داد. اما با اضافه شدن لایه‌های بیشتر، نه تنها آموزش داده‌ها مشکل‌تر شد، بلکه دقت شبکه و نتایج آن نیز اشباع و حتی تنزیل پیدا کرد.



شکل (1-1) نمونه‌ای از یک شبکه عمیق

## 1-2- معرفی شبکه‌های کانالوشنی - CNN

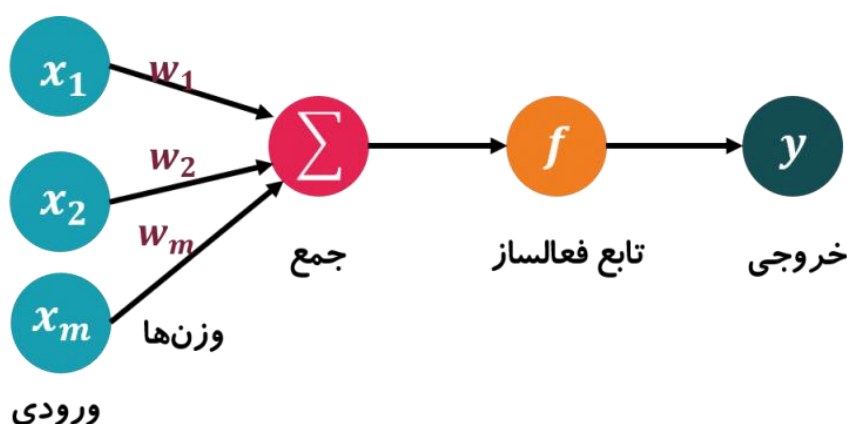
شبکه عصبی کانولوشن (Convolutional Neural Network) گل سرسبد ایده‌ها در یادگیری عمیق بوده است. به شبکه کانولوشن به اختصار CNN یا ConvNet گفته می‌شود. شبکه CNN در سال 1990 با الهام‌گیری از آزمایش‌های انجام‌شده توسط Hubel و Wiesel روی قشر بینایی (Visual Cortex) معرفی شد.

سرعت رشد شبکه کانولوشنی آنقدر زیاد بود که در مدت کوتاهی، در بسیاری از زمینه‌های مشکل بینایی کامپیوتر مانند شناسایی عمل انسان، تشخیص اشیاء، شناسایی چهره و ردیابی انقلابی برپا کرد. با سیطره بر بینایی کامپیوتر، شبکه کانولوشنی در سایر زمینه‌های هوش مصنوعی مانند پردازش زبان و گفتار نیز وارد شد.

## 1-2-1- عملکرد شبکه عصبی کانالوشنی

شبکه عصبی کانالوشن همانند سایر شبکه های عصبی (مثلا شبکه عصبی MLP) از لایه های نورونی با وزن و بایاس با قابلیت یادگیری تشکیل شده است. قطعا می دانید که در هر نورون اتفاقات زیر رخ می دهد:

- نورون مجموعه ای ورودی دریافت می کند.
- ضرب داخلی بین وزن های نورون و ورودی ها انجام می شود.
- حاصل با بایاس جمع می شود.
- در نهایت، از یک تابع غیر خطی (همان activation function) عبور داده می شود.



شکل (1-2) شماتیک نورون مصنوعی

اما تفاوت شبکه عصبی CNN و MLP در ورودی سیستم است. به عنوان مثال تصاویر. تصاویر معمولاً به شکل یک ماتریس دوبعدی از اعداد نمایش داده می شوند. هر درایه در این ماتریس دوبعدی معادل با یک پیکسل هست. اگر یک تصویر  $100 \times 100$  داشته باشیم، یعنی 10000 پیکسل داریم که به صورت دوبعدی کنار هم قرار گرفته اند.

حال تصور کنید بخواهیم یک لایه ورودی برای این 10000 پیکسل بسازیم؛ باید 10000 نورون برای لایه ورودی شبکه MLP در نظر بگیریم. اضافه کردن نورون و لایه بیشتر به این شبکه MLP باعث می شود شبکه ما شامل حجم بزرگی از پارامترها شود، محاسباتش هزینه بر باشد و البته Overfitting اتفاق بیفتد.

اما در شبکه های کانالوشنی اینگونه نیست و این شبکه ها به گونه ای طراحی شده اند که برای ورودی های با ساختار ماتریسی (دوبعدی و سه بعدی) به خوبی کار می کنند. شبکه MLP، ساختار داده های ورودی را عوض می کند و یک ماتریس دوبعدی  $100 \times 100$  را تبدیل به یک بردار به ابعاد 10000 می کند. اما شبکه CNN ساختار ورودی را عوض نمی کند و به ارتباط بین پیکسل های همسایه اهمیت می دهد.

اما این پرسش مطرح می شود که اساساً ارتباط بین پیکسل ها چرا هائز اهمیت است؟ ارتباط پیکسل هاست که یک تصویر را تشکیل می دهد.

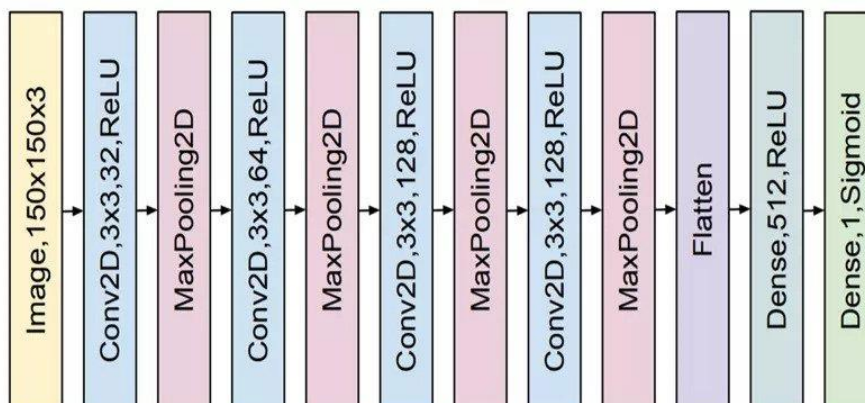
بنابراین، مهم است که ساختار تصویر یا ورودی را عوض نکنیم (تبدیل به بردار نکنیم) و همان ورودی اصلی را به شبکه بدهیم. دقیقاً همان کاری که شبکه CNN انجام می دهد. اما عملکرد CNN به داده های دوبعدی خلاصه نمی شود بلکه داده های زیر را نیز پشتیبانی می کند:

- داده‌های یک بعدی: سیگنال و sequence (مثلا یک sequence از کلمات)
- داده‌های دوبعدی: تصویر و طیف (spectrogram) صوت
- داده‌های سه بعدی: ویدئو و تصاویر حجمی (مثلا تصاویر MRI)
- داده‌های چهاربعدی: تصاویر حجمی همراه با زمان (مانند fMRI)

## 1-2-2- یک معماری ساده از معماری شبکه عصبی کانولوشن

در شکل (1-3) یک مثال ساده از معماری شبکه عصبی کانولوشن نشان داده شده است. شبکه کانولوشن معمولاً از بلوک‌های مختلفی تشکیل شده است. لایه‌ها یا بلوک‌های مختلف در شبکه CNN عبارتند از:

- لایه ورودی (Input layer)
- لایه کانولوشن (Convolutional layer)
- لایه غیرخطی (Non-linear activation function) (معمولاً تابع غیرخطی را همراه با لایه کانولوشنی یک‌جا نشان می‌دهند)
- لایه پولینگ (Pooling layer)
- لایه فولی کانکتد (Fully connected layer)



شکل (1-3) معماری شبکه عصبی کانولوشن

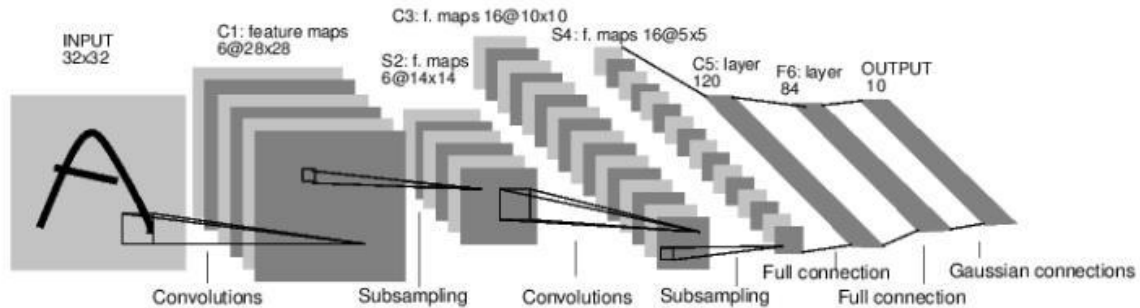
## 1-3- انواع شبکه‌های کانالوشنی

### LeNet -1-3-1

اولین شبکه ی کانولوشنالی که بسیار مطرح شد شبکه ی LeNet-5 بود که در مقاله ی "Gradient-based learning applied to document recognition" نوشته شده توسط LeCun, Bottou, Bengio, Haffner و در سال 1998 ارائه شد. نام شبکه نیز از نام Yann Lecun یکی از نویسندگان این مقاله برداشته شده است. این شبکه را می



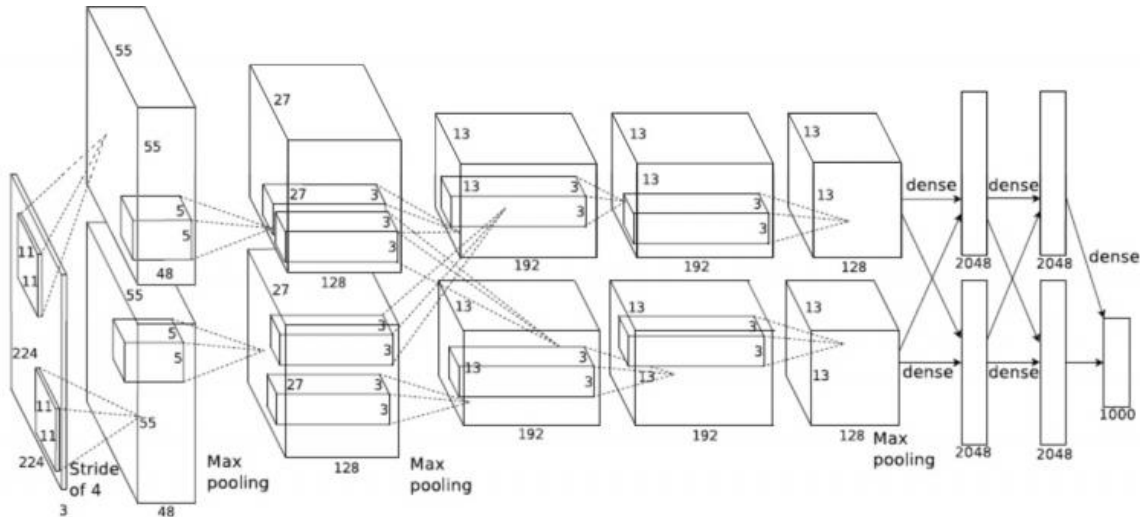
توان به عنوان اولین شبکه که فیلترهای کانولوشنالی در آن استفاده شده است نام برد. هدف این شبکه، خواندن کد پستی از پشت بسته های پستی بود.



شکل (1-4) LeNet

### AlexNet -1-3-2

دومین شبکه ی نورونی کانولوشنالی مهمی که می توان نام برد شبکه ی AlexNet است که توسط Alex Krizhevsky و حدود چهارده سال بعد از LeNet یعنی در سال 2012 معرفی شد. این شبکه دارای 8 لایه (پنج لایه ی کانولوشنالی و سه لایه fully connected) است که موجب شده در دسته ی شبکه های کم عمق (Shallow) قرار گیرد.

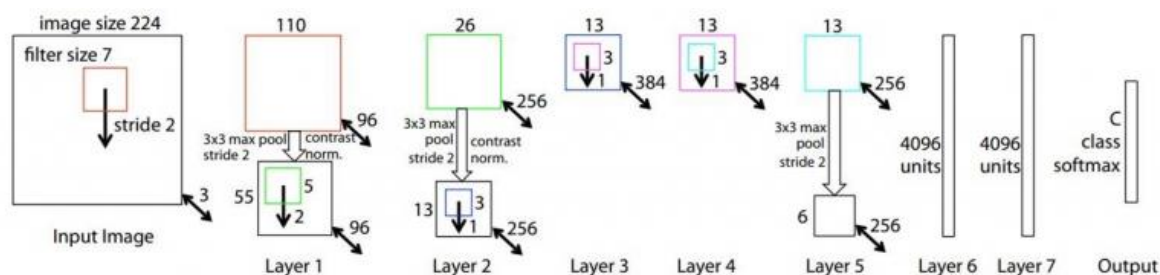


شکل (1-5) AlexNet

### ZFNET -3-3-1

نویسنده های این شبکه، Matthew Zeiler و Rob Fergus بودند که نام شبکه نیز از آن ها گرفته شده. این شبکه از ایده زیادی برخوردار نیست و مهم ترین ایده آن، قسمت Visualization آن است. ابزاری که با استفاده از آن، کاربرد هر نورون در شبکه تا حدی مشخص می شود و به درک شبکه های کانولوشنالی کمک بزرگی کرد. در این ابزار می توان دید که در لایه های نخست، شبکه خطوط و لبه های ساده را تشخیص می دهد و در لایه های بعد نورون ها مفاهیم پیچیده تری

مانند صورت انسان یا نوشته را تشخیص می‌دهند. بعضی از نورون‌ها برای ما قابل فهم نیستند. ولی معلوم شده حذف آن‌ها هم تاثیری مانند حذف دیگر نورون‌ها در عملکرد نهایی شبکه دارد.



شکل (1-6) ZFNet

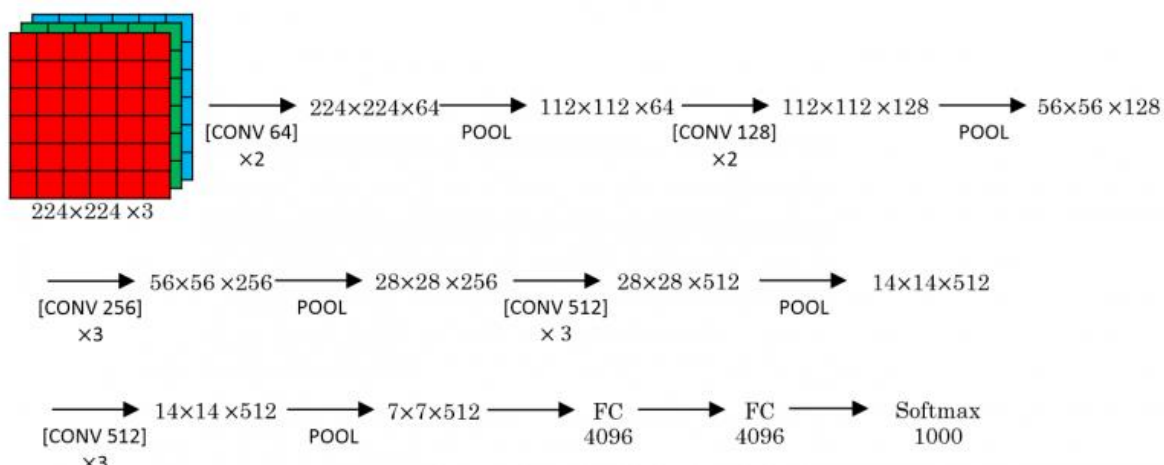
### VGG-4-3-1

این شبکه، به دلیل این که تعداد HyperParameter ها را کاهش داده، پیچیدگی کمتری نسبت به AlexNet و ZFNet دارد و به دلیل همین سادگی، از محبوبیت بالایی برخوردار است. در VGG در هر جا:

CONV به معنی فیلتر  $3 \times 3$  با stride 3 و same است.

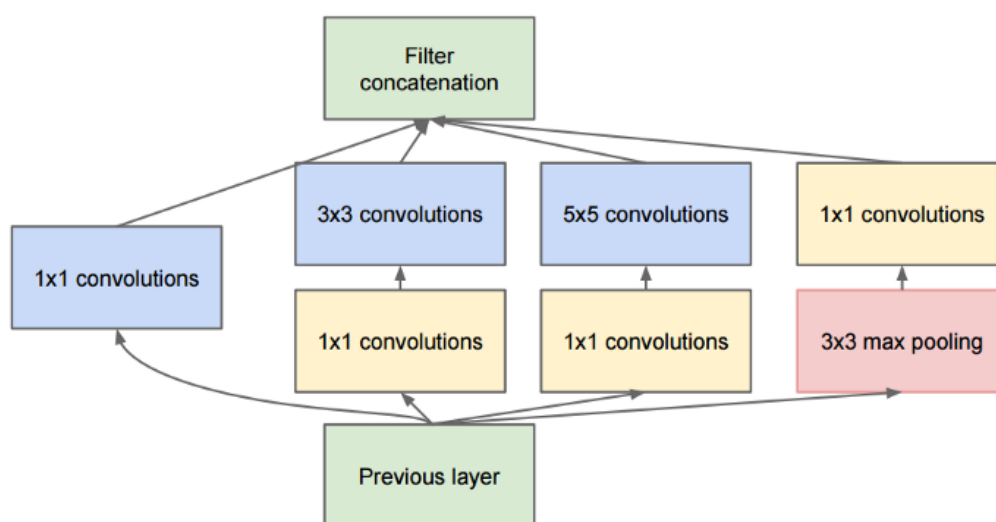
و POOL به معنی پولینگ  $2 \times 2$  با stride 2 است.

سایز تصویر ورودی  $224 \times 224 \times 3$  است. سپس دو بار CONV با عمق 64 روی آن اعمال می‌شود. (با CONV اول، تصویر  $224 \times 224 \times 64$  می‌شود و با CONV دوم دوباره  $224 \times 224 \times 64$  می‌شود). سپس POOL روی آن اعمال می‌شود و تصویر  $112 \times 112 \times 64$  می‌شود. (پولینگ روی عمق تاثیر نمی‌گذارد). بقیه مراحل در شکل زیر قابل مشاهده هستند:



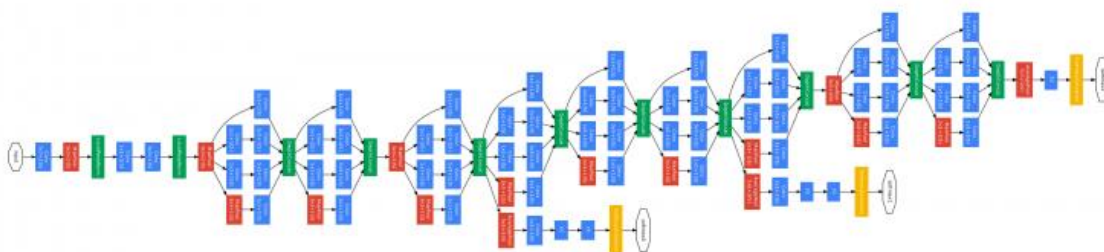
## GoogLeNet (Inception) -5-3-1

این شبکه توسط Google در سال 2014 ارائه شد. نام آن ادای احترامی به Yann Lecun، سازنده LeNet، اولین شبکه کانولوشنالی، است. این شبکه با ایده عمیق تر کردن شبکه های کانولوشنالی که طبق گفته سازنده، برگرفته از دیالوگی در فیلم Inception است! برای همین به این نام نیز شناخته می شود. در این شبکه با این رویکرد که هایپر پارامتر های شبکه را هم به گونه ای به پارامتر قابل یادگیری تبدیل کنند، ماژولی به نام Inception module طراحی شد.



شکل (1-7) GoogLeNet (Inception)

در این ماژول، ۳ فیلتر کانولوشنالی با اندازه های متفاوت و یک فیلتر پولینگ روی لایه قبلی اجرا شده و نتایج را به صورت یک تنسور واحد کنار هم می گذارد. در این صورت شبکه می تواند در لایه بعدی تصمیم بگیرد که از نتیجه کدام فیلتر و به چه اندازه می خواهد استفاده کند. سایز فیلتر های کانولوشنالی به صورتی که در تصویر مشخص است،  $1 \times 1$ ،  $3 \times 3$  و  $5 \times 5$  است و فیلتر پولینگ با سایز  $3 \times 3$  روی آن اجرا شده.



شکل (1-8) نحوه قرارگیری فیلتر کانولوشنی

## فصل 2: بررسی شبکه Resnet

### 1-2- پیدایش شبکه عصبی باقی مانده‌ای

شبکه‌ی رزنت (ResNet) یا به صورت کامل، Residual Network یکی از شبکه‌های عمیق معروف است. این شبکه را شیو کینگ رن (Shaoqing Ren)، کیمینگ هی (Kaiming He)، ژان سان (Jian Sun) و زایانگیا ژوئنگ (Xiangyu Zhang) در سال ۲۰۱۵ معرفی کردند. مدل رزنت تاکنون یکی از محبوب‌ترین و موفق‌ترین مدل‌های یادگیری عمیق بوده است. این مدل برنده‌ی چالش ILSVRC در سال ۲۰۱۵ بود. دلیل موفقیت شبکه‌ی رزنت (ResNet) این است که به ما امکان آموزش شبکه‌های عصبی بسیار عمیق با بیش از ۱۵۰ لایه را داد. قبل از رزنت (ResNet) شبکه‌های عصبی بسیار عمیق، به دلیل مشکل محوشدگی گرادیان (Vanishing Gradient)، دچار مشکل می شدند.

### 2-2- مزیت‌های شبکه Resnet

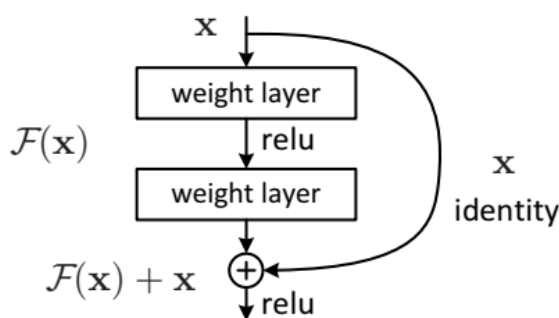
شبکه‌های یادگیری عمیق معمولی، مانند AlexNet، ZFNet و VGGNet، اغلب لایه‌های کانولوشنی و سپس لایه‌های کاملاً متصل (Fully Connected) برای طبقه‌بندی دارند، بدون هیچ گونه اتصال میان‌بر. ما در اینجا آن‌ها را شبکه‌های ساده (Plain Networks) می نامیم.

وقتی شبکه‌ی ساده (Plain Networks) عمیق تر هستند (یعنی لایه‌ها افزایش می یابند)، مشکل محوشدگی گرادیان (Vanishing Gradient) یا انفجار گرادیان (Exploding Gradient) رخ می دهد؛ بنابراین عمیق تر کردن شبکه کار راحتی محسوب نمی شد که تنها با اضافه کردن لایه به شبکه آن را عمیق تر کنیم. اینجا بود که شبکه‌ی رزنت (ResNet) معرفی شد تا این مشکل را حل کند. این شبکه می تواند تا ۱۵۲ لایه داشته باشد.

#### 1-2-2-1- اتصالات میان‌بر، نقطه قوت رزنت

اتصالات میان‌بر (Skip Connections) یا اتصالات اضافی (Residual Connections) راه حلی بود که شبکه رزنت (ResNet) برای حل مشکل شبکه‌های عمیق ارائه کرد.

در شکل (1-2) یک بلاک اضافی (Residual Block) را مشاهده می کنیم. همان طور که مشخص است، فرق این شبکه با شبکه‌های معمولی این است که یک اتصال میان‌بر دارد که از یک یا چند لایه عبور می کند و آن‌ها را در نظر نمی گیرد؛ درواقع به نوعی میان‌بر می زند و یک لایه را به لایه‌ی دورتر متصل می کند.



شکل (2-1) اتصال میانبر در زرنت

## 2-2-2- طرز کار شبکه زرنت با اتصالات میانبر

با توجه به اینکه اکنون یک اتصال اضافی داریم، بنابراین خروجی این بلاک دیگر مانند قبل نیست؛ درواقع قبل از اضافه شدن این اتصال مقدار ورودی  $x$  در وزن متناظرش ضرب می‌شد و با مقدار بایاس (Bias) جمع می‌شد و در نهایت یک تابع فعال‌ساز ReLU روی آن اعمال می‌شد.

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

یا به عبارت دیگر:

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

اما اکنون که اتصال میانبر را نیز داریم، مقدار  $x$  اتصال میانبر هم به مقدار  $F(x)$  اضافه می‌شود.

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

اما اینجا یک مشکل وجود دارد؛ از آنجا که در طول شبکه‌ی خروجی لایه‌های مختلف کانولوشن ابعاد مختلفی را دارد، بنابراین ممکن است مقدار  $x$  که اتصال میانبر از لایه‌های قبلی به  $f(x)$  اضافه می‌کند ابعاد متفاوتی را رقم بزند. برای حل این مشکل دو راه حل وجود دارد:

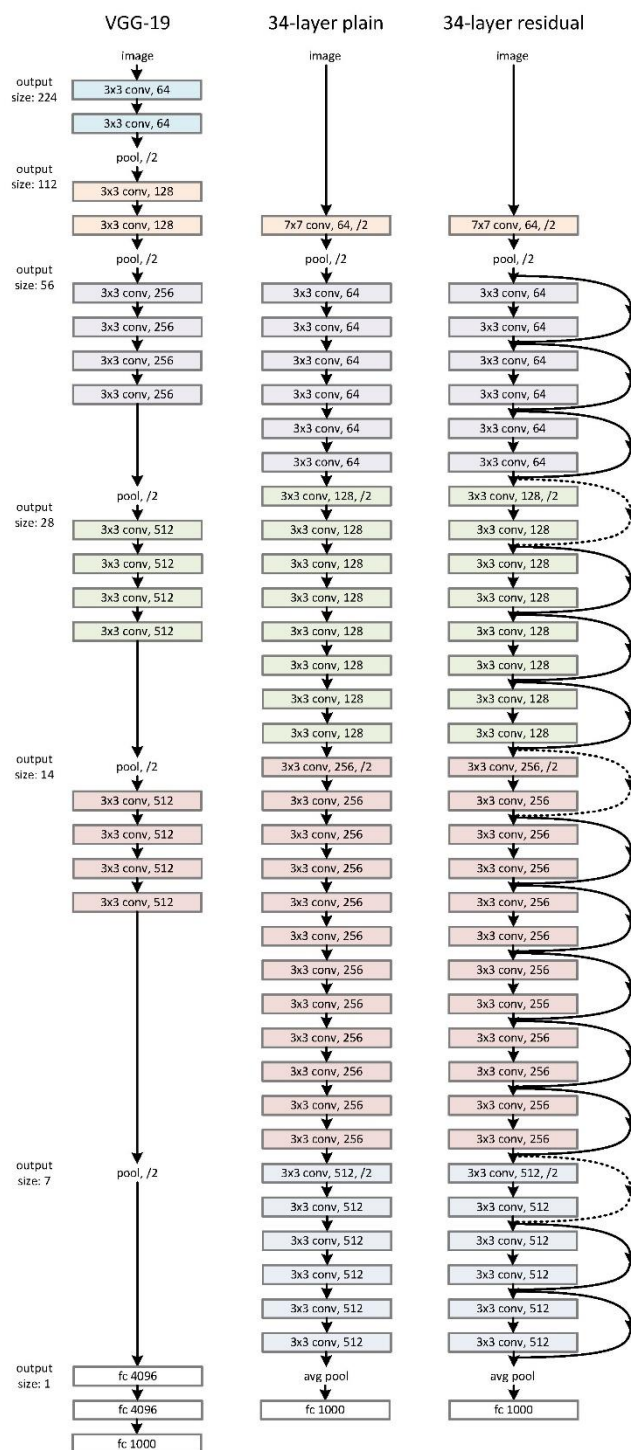
ابعاد اتصال میانبر را با استفاده از فرایند لایه‌گذاری با صفر (Zero padding) افزایش می‌دهیم و از لایه‌های کانولوشن  $1 \times 1$  برای کاهش ابعاد ورودی استفاده می‌کنیم؛ در این حالت مقدار خروجی به این صورت محاسبه می‌شود:

$$H(x) = f(x) + w1.x$$

در اینجا ما یک پارامتر (وزن)  $w1$  به مقدار  $x$  اضافه می‌کنیم. البته در راه حل اول این اتفاق نمی‌افتد؛ درواقع با استفاده از اتصالات میانبر در هنگام انتشار روبه عقب، اگر مقدار گرادیان خیلی کوچک شود، این امکان را داریم که با استفاده از این اتصالات به لایه‌های اولیه برسیم.

## 3-2-2- معماری شبکه‌ی رزنت (ResNet)

رزنت از یک شبکه‌ی ۳۴ لایه‌ای ساده استفاده می‌کند که از معماری VGGNet الهام گرفته شده و به این شبکه‌ی اتصالات میان‌بر اضافه شده است. این شکل نمایی از معماری رزنت (ResNet) را نشان می‌دهد.

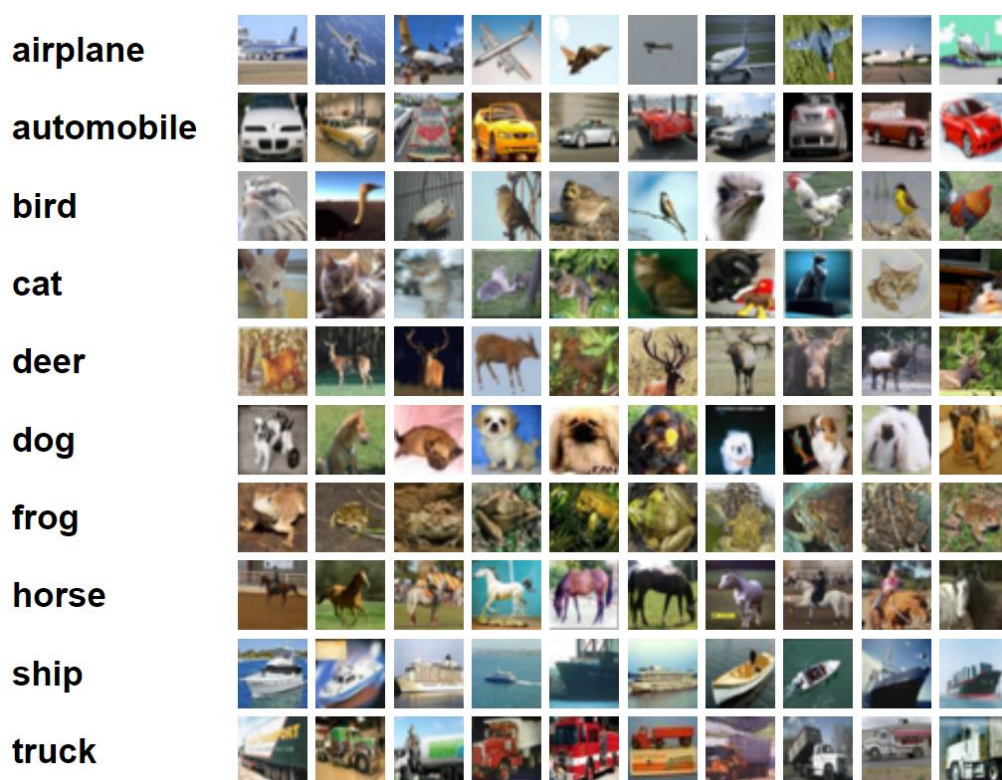


شکل (2-2)

## فصل 3: پایاده‌سازی شبکه رزنت برای طبقه‌بندی تصاویر دیتاست CIFAR-10

### 3-1- دیتاست CIFAR-10

در این پروژه با مسئله‌ی classification ده کلاسه، برای دسته‌بندی اشیاء مختلف موجود در دیتاست CIFAR-10 روبه‌رو بودیم. این دیتاست شامل 60 هزار تصویر با اندازه‌ی 32 در 32 پیکسل است که در ده گروه مختلف دسته‌بندی می‌شوند. از این 60 هزار تصویر، 10 هزار تصویر برای تست نهایی در نظر گرفته شده است. برخی از تصاویر این دیتاست در شکل (3-1) آورده شده است.



شکل (3-1) برخی از تصاویر موجود در دیتاست CIFAR-10 به همراه کلاس آن‌ها

### 3-2- پیش پردازش داده‌ها

برای پیش‌پردازش داده‌ها، ابتدا لیبل‌های تصاویر را که اعدادی صحیح بین 0 تا 10 هستند، به صورت one-hot کدگذاری نمودیم؛ این مرحله برای سازگاری داده‌ها با شبکه‌ی عصبی لازم است. جهت جلوگیری از overfit شدن مدل، بر روی داده‌های آموزش، به صورت تصادفی تبدیل‌های هندسی و رنگی را

اعمال نمودیم. این تبدیل‌ها عبارتند از:

- Random Flipping
- Random Saturation
- Random Contrast
- Random Brightness

بجز موارد بالا، به کمک ماژول applications کتابخانه‌ی keras، تصاویر متناسب با شبکه‌ی ResNet پیش‌پردازش شدند. همچنین سائز تصاویر، به کمک تابع UpSampling2D کتابخانه‌ی keras، به 256 در 256 پیکسل افزایش یافت؛ چرا که ورودی مناسب برای شبکه‌ی ResNet در حدود 220 در 220 پیکسل است و تصاویر با سائز 32 در 32 برای آموزش توسط این شبکه بسیار کوچک هستند.

### 3-3- ساختار مدل

برای ایجاد مدل، از مدل از پیش آموزش یافته‌ی ResNet50 بر روی دیتاست imagenet استفاده نمودیم؛ به شکلی که پس از لایه‌های کانولوشنی، یک لایه‌ی GlobalMaxPooling2D برای تبدیل خروجی کانولوشنی به بردار قرار گرفت. در لایه‌ی بعد از آن، برای نرمالیزاسیون مقادیر خروجی، از لایه‌ی BatchNormalization استفاده نمودیم. سپس یک لایه‌ی Dropout با نرخ 0.3 برای جلوگیری از overfit شدن مدل، و یک لایه‌ی Dense با 10 نورون و تابع فعال‌ساز softmax برای طبقه‌بندی نهایی تصاویر قرار گرفت. ساختار کلی لایه‌ها در شکل (3-2) نشان داده شده است.

Model: "ResNet50"

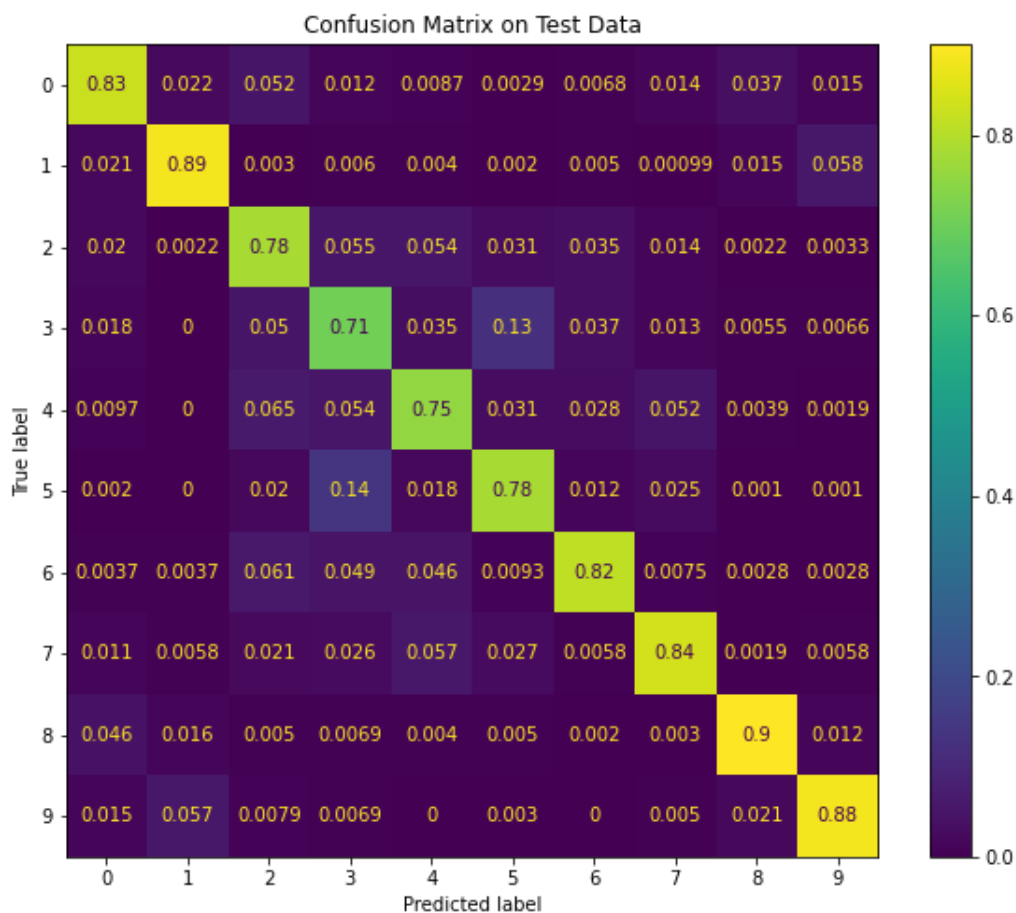
Layer (type)	Output Shape	Param #
Input_Layer (InputLayer)	[(None, 32, 32, 3)]	0
Resize_1 (UpSampling2D)	(None, 64, 64, 3)	0
Resize_2 (UpSampling2D)	(None, 128, 128, 3)	0
Resize_3 (UpSampling2D)	(None, 256, 256, 3)	0
tf.__operators__.getitem_4 (SlicingOpLambda)	(None, 256, 256, 3)	0
tf.nn.bias_add_4 (TFOpLambda)	(None, 256, 256, 3)	0
resnet50 (Functional)	(None, 8, 8, 2048)	23587712
Max_Pool (GlobalMaxPooling2D)	(None, 2048)	0
Batch_Norm (BatchNormalization)	(None, 2048)	8192
Dropout (Dropout)	(None, 2048)	0
Classifier (Dense)	(None, 10)	20490
=====		
Total params: 23,616,394		
Trainable params: 24,586		
Non-trainable params: 23,591,808		

شکل (3-2)



### 3-4- آموزش مدل

برای آموزش مدل، وزن های لایه های از پیش آموزش داده شده ی کانولوشنی را ثابت نگه داشتیم و تنها وزن های لایه های BatchNormalization و Dense را آموزش دادیم. برای آموزش از بهینه ساز Stochastic Gradient Descent با نرخ یادگیری 0.01 و تابع هزینه ی Categorical Cross Entropy و Batch Size برابر با 64 استفاده نمودیم. پس از 6 مرحله آموزش، میانگین دقت داده های تست به 82 درصد رسید. شکل (3-3) ماتریس درهم زنی نرمالیزه شده ی مدل را بر روی داده های تست نشان می دهد.



شکل (3-3)

طبق شکل (3-3)، می بینیم که دقت مدل بر روی همه ی کلاس ها، بین 71 تا 90 درصد است. طبق این ماتریس می توانیم ببینیم که 13 درصد داده های کلاس 3 اشتباه در کلاس 5 قرار گرفته است و 14 درصد داده های کلاس 5 در کلاس 3 قرار گرفته است. کلاس های 3 و 5 به ترتیب مربوط به گربه و سگ است. با توجه به این که سائز تصاویر اصلی 32 در 32 بوده است و ویژگی های ظاهری سگ و گربه بسیار به هم نزدیک هستند، خطای مدل در این مورد قابل توجیه است.

## مراجع

- [1] Deep Residual Learning for Image Recognition Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Su  
<https://arxiv.org/abs/1512.03385>