



آکادمی هوش مصنوعی

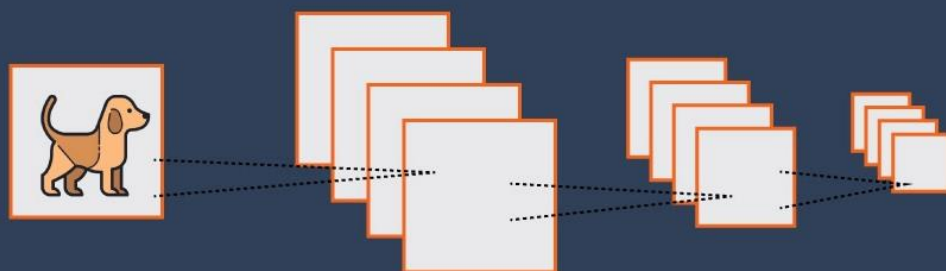
هوسم | howsam.org

شبکه عصبی کانولوشن

در این پست می‌خواهم به آموزش شبکه عصبی کانولوشن بپردازم. این پست، یکی از سلسله پست‌های **آموزش یادگیری عمیق رایگان** است. در این آموزش شبکه کانولوشن را برای موضوع طبقه‌بندی (Classification) توضیح داده‌ام. امیدوارم با این آموزش دوستی خوبی بین شما و شبکه کانولوشنی شکل بگیرد. با هوسم همراه باشید...

با احترام و ❤️ :

سید سجاد اشرفی



دوره یادگیری عمیق رایگان هوسم

آموزش شبکه عصبی کانولوشن، یک جلسه از مجموعه جلسات دوره یادگیری عمیق رایگان هوسم است. لینک سایر جلسات دوره یادگیری عمیق را در زیر می‌توانید مشاهده کنید.

لینک	لیست جلسات
کلیک کنید	معرفی دوره یادگیری عمیق رایگان
کلیک کنید	جلسه 01: ساختار نورون مصنوعی
کلیک کنید	جلسه 02: یادگیری نورون مصنوعی
کلیک کنید	جلسه 03: شبکه عصبی MLP
کلیک کنید	جلسه 04: دسته بندی با MLP و تنسورفلو
کلیک کنید	جلسه 05: رگرسیون با MLP و تنسورفلو
کلیک کنید	جلسه 06: شبکه عصبی کانولوشنی (CNN)
کلیک کنید	جلسه 07: شبکه عصبی بازگشتی (RNN)
کلیک کنید	جلسه 08: شبکه عصبی GAN
کلیک کنید	جلسه 09: شبکه عصبی LSTM
کلیک کنید	جلسه 10: شبکه عصبی موبایل‌نت

فهرست مطالب

- ۱- شبکه عصبی کانولوشن چیست؟ ۱
- ۲- آشنایی با شبکه عصبی کانولوشن ۳
- ۳- تصویر در پردازش تصویر ۷
- ۴- کانولوشن چیست؟ ۱۰
- ۵- لایه کانولوشن در شبکه عصبی کانولوشن (Convolutional Layer) ۱۶
- ۶- لایه غیرخطی در شبکه عصبی کانولوشن (Activation Function) ۲۱
- ۷- لایه پولینگ در شبکه عصبی کانولوشن (Pooling Layer) ۲۲
- ۸- طراحی شبکه عصبی کانولوشن ساده ۲۴
- ۹- لایه فولی کانکتد در شبکه کانولوشن (Fully Connected Layer) ۲۵
- ۱۰- سوالات رایج درباره شبکه عصبی کانولوشن ۲۵
- ۱۱- واقعا به پایان رسیدم؟! ۲۶

۱- شبکه عصبی کانولوشن چیست؟

از زمان پیدایش یادگیری عمیق (Deep Learning)، 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 ۱۹۹۰ با الهام‌گیری از آزمایش‌های انجام‌شده توسط Hubel و Wiesel روی قشر بینایی (Visual Cortex) معرفی شد. یکی از اولین پروژه‌های انجام‌شده با این شبکه CNN، پروژه معروف شناسایی ارقام دست‌نویس MNIST در سال ۱۹۹۸ توسط Yann Lecun بود که نتایج امیدوارکننده‌ای به همراه داشت.



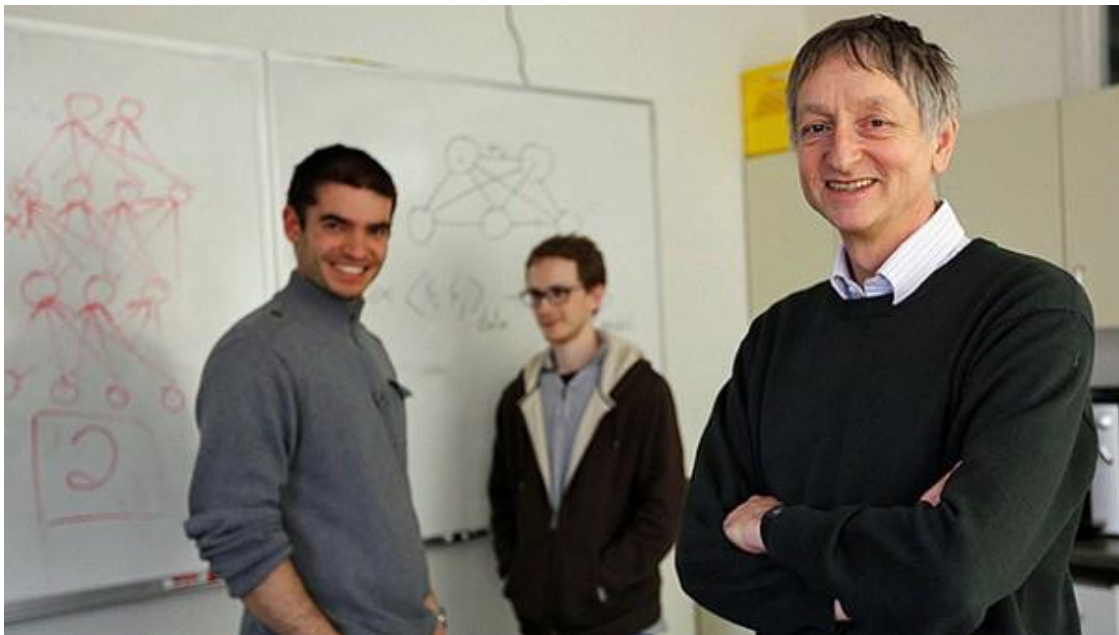
شکل ۱: Hubel و Wiesel که در سال ۱۹۶۲ آزمایش‌هایی روی قشر بینایی (Visual Cortex) گریه انجام دادند و در دهه ۹۰ شبکه کانولوشن را ارائه کردند.

۱-۱- گرد و خاک به‌پاکردن شبکه عصبی کانولوشن

اما درخشش اصلی شبکه کانولوشن در سال ۲۰۱۲ در جریان رویداد "ImageNet Large Scale Recognition Challenge" (همان مسابقه ImageNet) اتفاق افتاد. اتفاقی که توجه جامعه محققین را به خود جلب کرد. در مسابقه ImageNet سال ۲۰۱۲، Alex Krizhevsky از دانشگاه تورنتو با شبکه AlexNet وارد مسابقه پیچیده ImageNet شد. مسابقه ImageNet یک چالش دسته‌بندی (Classification) با ۱.۲ میلیون تصویر در ۱۰۰۰ کلاس

بود. شبکه AlexNet با خطای ۱۶.۴٪ به رتبه اول مسابقه رسید. به نظر شما نفر دوم به چه خطایی دست یافته بود؟ ۲۶.۲٪! اختلاف خطای حداقل ۱۰ درصدی بین شبکه CNN با سایر روش‌ها...

با این نتیجه، AlexNet مشتمل بر یک کوبید و نقشه راه جدیدی در بینایی کامپیوتر ترسیم کرد. از آن زمان تا امروز (۲۰۲۰)، تحقیقات CNN رشد چشم‌گیری داشته است. تنها در ۳ سال، محققان از ۸ لایه AlexNet به ۱۵۲ لایه ResNet رسیدند و میزان خطای چالش ImageNet را به کمتر از ۴٪ کاهش دادند. ۱۲٪ کاهش خطا طی ۳ سال با شبکه CNN، البته با افزایش پیچیدگی ساختار شبکه CNN!



شکل ۲: گروهی که روی شبکه AlexNet کار کردند. راست: Geoffrey Hinton، وسط: Alex Krizhevsky، چپ: Ilya Sutskever. آقای Alex Krizhevsky حواسش به دوربین نیست و دل در گروی نوروں یار (AlexNet) بسته!

۱-۲- طوفان شبکه کانولوشن

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 و اتفاقاً نتایج خوبی هم بدست آورد. مثلاً، گوگل با استفاده از شبکه کانولوشن در یادگیری تقویتی، مدل AlphaGo را ساخت که توانست بهترین بازیکن بازی سخت Go را قدرتمندانه شکست دهد و در سرتیتر اخبار قرار گیرد. شبکه عصبی کانولوشن همچون مارکوپولو وارد قلمروهای مختلف شد و حتی در حوزه پزشکی نیز به

دستاوردهای قابل اعتنایی دست یافت. با این دستاوردها، شبکه CNN فرزند خلف یادگیری عمیق بود که باعث شد یادگیری عمیق به شهرتی روزافزون دست یابد.



شکل ۳: مسابقه GO بین Lee Sedol (سمت راست) و AlphaGo؛ اون آقای سمت چپی AlphaGo نیست و صرفاً حرکات رو براساس اونچه که در مانیتور نشون داده میشه، اجرا میکنه!

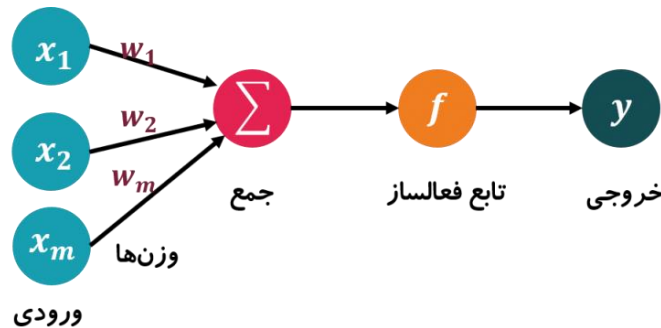
بعد از جواب به سوال "شبکه عصبی کانولوشن چیست؟"، می‌خواهم کمی شما را با شبکه کانولوشن آشنا کنم. در بخش بعدی، خیلی وارد جزئیات نمی‌شوم، اما رفته رفته تا آخر این پست به جزئیات شبکه CNN هم خواهیم رسید.

۲- آشنایی با شبکه عصبی کانولوشن

شبکه عصبی کانولوشن همانند سایر شبکه های عصبی (مثلا شبکه عصبی MLP) از لایه‌های نورونی با وزن و بایاس با قابلیت یادگیری تشکیل شده است. قطعاً می‌دانید که در هر نورون اتفاقات زیر رخ می‌دهد:

- ۱- نورون مجموعه‌ای ورودی دریافت می‌کند.
- ۲- ضرب داخلی بین وزن‌های نورون و ورودی‌ها انجام می‌شود.
- ۳- حاصل با بایاس جمع می‌شود.
- ۴- درنهایت، از یک تابع غیرخطی (همان activation function) عبور داده می‌شود.

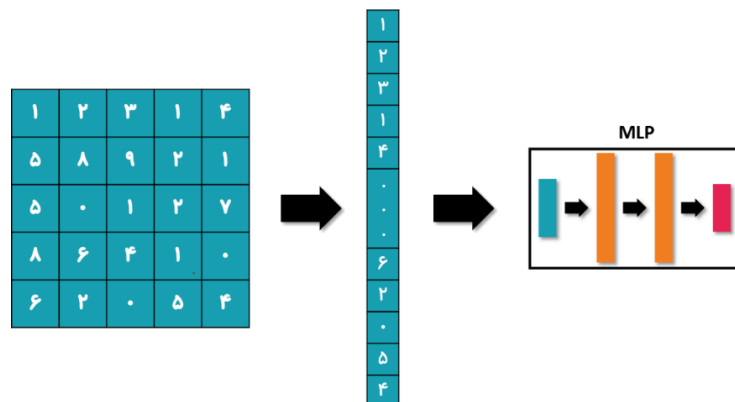
فرآیند بالا لایه به لایه انجام می‌شود و در نهایت به لایه خروجی می‌رسیم. لایه خروجی، پیش‌بینی شبکه را تولید می‌کند. اما با این حد از تشابه بین شبکه MLP و شبکه CNN، پس تفاوت در کجاست؟ در ادامه به این سوال مهم جواب می‌دهم...



شکل ۴: شماییک نورون مصنوعی (artificial neuron).

۲-۱- تفاوت شبکه عصبی MLP و کانولوشن

تفاوت در ورودی است، مثلاً تصویر... تصاویر معمولاً به شکل یک ماتریس دوبعدی از اعداد نمایش داده می‌شوند. هر درایه در این ماتریس دوبعدی معادل با یک پیکسل هست. در بخش بعدی درمورد تصویر بیشتر توضیح خواهم داد. اگر یک تصویر 100×100 داشته باشیم، یعنی ۱۰۰۰۰ پیکسل داریم که به صورت دوبعدی به صورت مسالمت‌آمیز کنار هم‌دیگر نشسته‌اند. حال تصور کنید بخواهیم یک لایه ورودی برای این ۱۰۰۰۰ پیکسل بسازیم؛ باید ۱۰۰۰۰ نورون برای لایه ورودی شبکه MLP در نظر بگیریم. اضافه کردن نورون و لایه بیشتر به این شبکه MLP باعث می‌شود شبکه ما شامل حجم بزرگی از پارامترها شود، محاسباتش هزینه‌بر باشد و البته Overfitting اتفاق بیفتد.



شکل ۵: اعمال تصویر ورودی به شبکه MLP؛ ابتدا تصویر باید به یک بردار تبدیل شود و بعد به شبکه MLP داده شود.

۲-۲- حالا نقطه قوت شبکه CNN کجاست؟

این شبکه‌ها به گونه‌ای طراحی شده‌اند که برای ورودی‌های با ساختار ماتریسی (دوبعدی و سه‌بعدی) به خوبی کار می‌کنند. شبکه MLP، ساختار داده‌های ورودی را عوض می‌کند و یک ماتریس دوبعدی 100×100 را تبدیل به یک بردار به ابعاد ۱۰۰۰۰ می‌کند. اما شبکه CNN ساختار ورودی را عوض نمی‌کند و به ارتباط بین پیکسل‌های همسایه اهمیت می‌دهد.

چرا باید به ارتباط بین پیکسل‌های همسایه اهمیت دهیم؟ به تصویر زیر نگاه کنید؛ مگر نه اینکه از کنارهم قرار گرفتن تعدادی پیکسل در راستای سطر و ستون یک تصویر از پاندای خسته تشکیل شده است؟ 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 می‌دهد.



شکل ۶: یک شی در تصویر از مجموعه‌ای پیکسل در کنار هم تشکیل می‌شود. بنابراین، حفظ این آرایش پیکسلی مهم است.

آیا شبکه CNN تنها برای تصویر دوبعدی مناسب است؟ خیر، داده‌های زیر غذای خوشمزه‌ای برای شبکه CNN هستند:

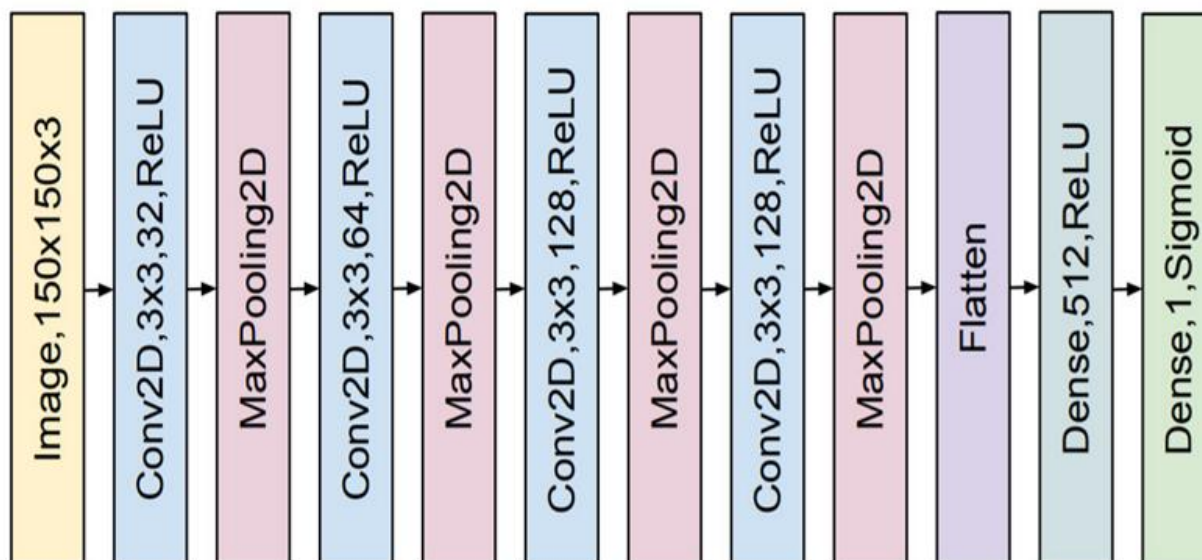
- داده‌های یک بعدی: سیگنال و sequence (مثلا یک sequence از کلمات)
- داده‌های دوبعدی: تصویر و طیف (spectrogram) صوت
- داده‌های سه بعدی: ویدئو و تصاویر حجمی (مثلا تصاویر MRI)
- داده‌های چهاربعدی: تصاویر حجمی همراه با زمان (مانند fMRI)

بعد از آشنایی با مزایای شبکه کانولوشن و بررسی تفاوتش با شبکه MLP، وقتش رسیده که معماری این شبکه را زیر تیغ جراحی ببریم. اما بیایید اول یک آشنایی کلی با معماری شبکه CNN بدست آوریم.

۲-۳- یک معماری ساده از معماری شبکه عصبی کانولوشن

در شکل ۷ یک مثال ساده از معماری شبکه عصبی کانولوشن نشان داده شده است. شبکه کانولوشن معمولا از بلوک‌های مختلفی تشکیل شده است. لایه‌ها یا بلوک‌های مختلف در شبکه CNN عبارتند از:

- لایه ورودی (Input layer) [بلوک زرد در شکل ۷]
- لایه کانولوشن (Convolutional layer) [بلوک‌های آبی در شکل ۷]
- لایه غیرخطی (Non-linear activation function) [بلوک‌های آبی در شکل ۷] (معمولا تابع غیرخطی را همراه با لایه کانولوشنی یک‌جا نشان می‌دهند)
- لایه پولینگ (Pooling layer) [بلوک‌های قرمز در شکل ۷]
- لایه فولی کانکتد (Fully connected layer) [بلوک‌های سبز در شکل ۷]



شکل ۷: یک مثال ساده از معماری شبکه عصبی کانولوشن.

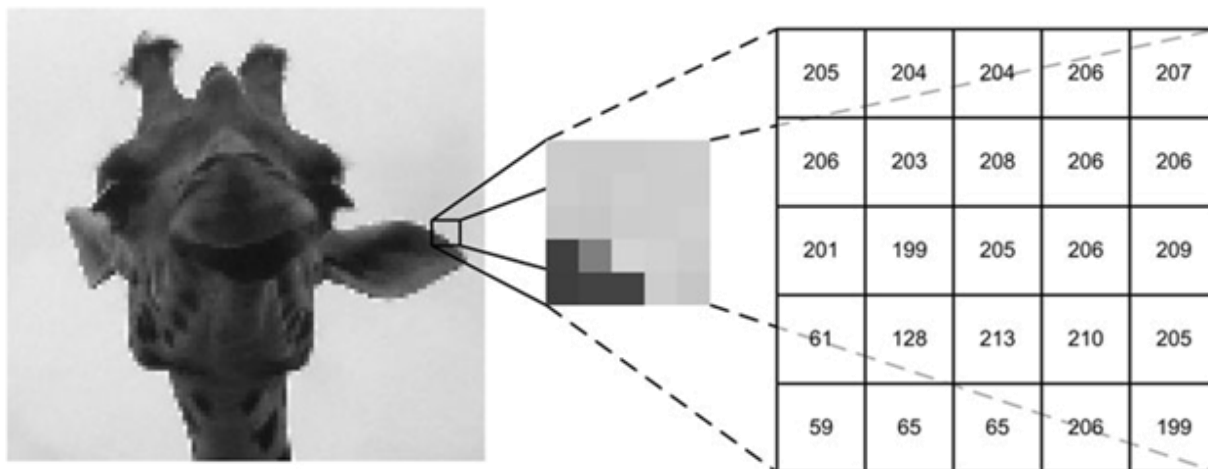
در ادامه این پست آموزش شبکه عصبی کانولوشن، لایه‌های بالا را توضیح می‌دهم. شاید بگویید که لایه ورودی که صرفاً تصویر است و جز لایه‌های شبکه محسوب نمی‌شود. اما، من لایه ورودی را هم اضافه کردم و لازم هست درباره آن توضیح دهم. پس، ابتدا ورودی را بررسی می‌کنیم، سپس به لایه‌های شبکه کانولوشن می‌رسیم. در بالا اشاره کردم که شبکه CNN ورودی‌های متنوعی را می‌پذیرد. اما من در اینجا می‌خواهم معروف‌ترین داده ورودی به شبکه CNN، یعنی تصویر را فقط بررسی کنم.

۳- تصویر در پردازش تصویر

در ساده‌ترین تعریف ممکن، تصویر یک آرایه یا ماتریس دوبعدی از اعداد است که هریک از درایه‌های آن معادل یک پیکسل هست. در شکل ۸ یک نمونه تصویر نشان داده شده است. نمایی زوم‌شده از تصویر نیز نشان داده شده. مشاهده می‌کنید که چگونه مربع‌های کوچک در کنارهم قرار گرفته‌اند. به جدول اعداد نگاه کنید؛ اگرچه در حالت معمولی ما یک تصویر با رنگ می‌بینیم (تصویر سمت چپ شکل ۸)، اما این اعداد هستند که این رنگ‌ها را می‌سازند (جدول سمت راست شکل ۸)!

مقادیر موجود در تصویر یا ماتریس بین ۰ تا ۲۵۵ هستند. مقدار صفر معادل با رنگ سیاه مطلق و مقدار ۲۵۵ معادل با رنگ سفید است. هرچقدر این مقادیر به صفر نزدیک‌تر باشد، آن پیکسل تیره‌تر می‌شود. هرچقدر مقدار پیکسل به ۲۵۵ نزدیک‌تر باشد، رنگ آن پیکسل روشن‌تر می‌شود. همان‌طور که در شکل ۸ مشاهده

می‌کنید، نواحی تیره در ناحیه بزرگنمایی شده (سمت چپ و پایین) مقادیر کمتری دارند (مثلا اعداد ۵۹ ۶۱ ۶۵). همچنین نواحی روشن (سمت راست و بالا)، مقادیر بزرگتری دارند (مثلا اعداد ۲۰۶ ۲۰۷ ۲۰۴).



شکل ۸: یک نمونه تصویر همراه نمایی زوم‌شده از تصویر؛ در ناحیه زوم‌شده، هم پیکسل‌ها را به راحتی می‌توانید مشاهده کنید و هم اعداد متناظر با هر پیکسل نشان داده شده است.

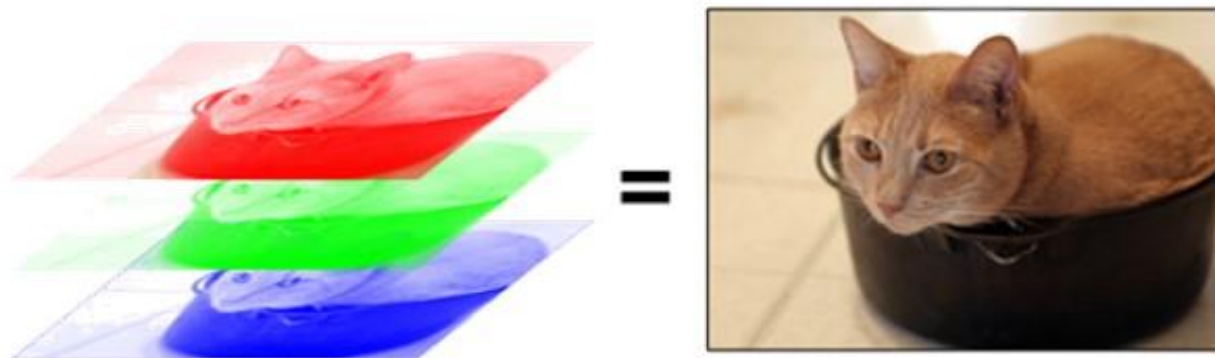
دو نوع تصویر سطح خاکستری (Gray scale image) و تصویر رنگی (RGB image) معمولا به عنوان ورودی به شبکه عصبی کانولوشن داده می‌شود.

۳-۱- تصویر سطح خاکستری در پردازش تصویر

ساده‌ترین نوع تصویر، یک تصویر سطح خاکستری است. در واقع این تصویر یک ماتریس دوبعدی است. تصویر شتر در شکل ۸ یک نمونه تصویر سطح خاکستری است. **شتر چیه؟! زرافه هست! 😊 داشتم متن رو بعد از مدتی مرور می‌کردم، دیدم شاخ شمشاد رو نوشتم شتر!**

۳-۲- تصویر رنگی در پردازش تصویر

تصاویر رنگی ساختار کمی متفاوت دارند. تصویر رنگی از سه صفحه تشکیل شده است. این صفحات عبارتند از صفحه قرمز (R)، صفحه سبز (G) و صفحه آبی (B). مشخصات هر صفحه مشابه همان تصویر سطح خاکستری است و مقادیر اعداد بین ۰ تا ۲۵۵ است. از ترکیب سه عدد، یک رنگ نهایی حاصل می‌شود. به همین خاطر به تصاویر رنگی، تصویر RGB گفته می‌شود. از ترکیب این سه صفحه، یک تصویر رنگی حاصل می‌شود. یک نمونه تصویر RGB در شکل ۹ نشان داده شده است.



شکل ۹: تصویر رنگی و سه صفحه قرمز، سبز و آبی. از ترکیب اعداد سه صفحه، تصویر سمت راست ساخته می‌شود.

پس تصویر رنگی از سه صفحه RGB تشکیل شده است. حالا که تصویر را شناختید، می‌خواهم شما را با تنسور آشنا کنم. در یادگیری عمیق به وفور کلمه تنسور را می‌شنوید. به صورت کوتاه در ادامه تنسور را توضیح می‌دهم.

۳-۳- تنسور چیست؟

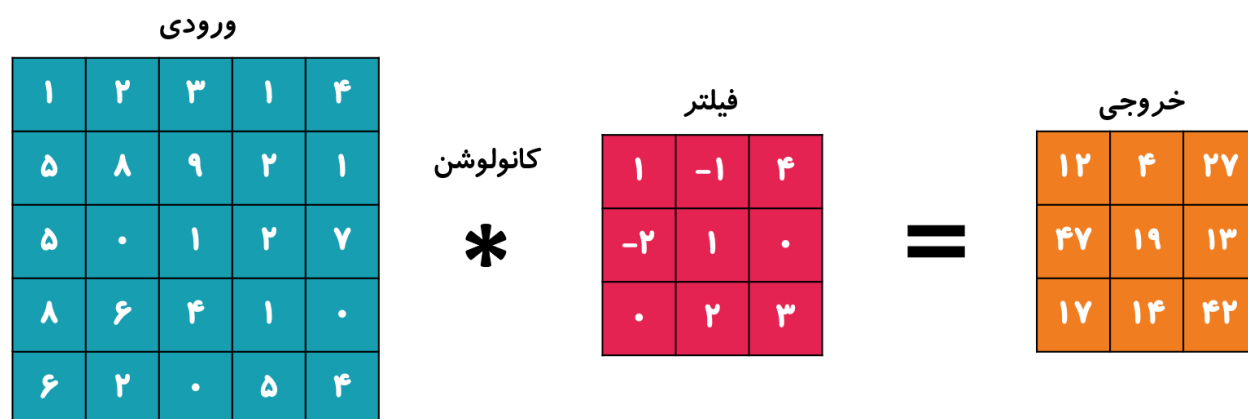
در یادگیری عمیق، عبارت تنسور را زیاد خواهیم شنید. در گذشته همه ما با گونه‌های خاصی از تنسورها کار کردیم. اعداد، بردارها و ماتریس‌ها همگی حالات خاص یک تنسور هستند! تنسورها در واقع تعمیم یافته ماتریس‌ها هستند و می‌توان آنها را به صورت یک آرایه چندبعدی نمایش داد. یعنی، به عدد، بردار و ماتریس به ترتیب تنسور صفربعدی، یک بعدی و دوبعدی گفته می‌شود. دقت کنید که تنسورها دقیقاً همانند ماتریس‌ها و بردارها می‌توانند اندیس‌دهی شوند. یعنی ما با کمک اندیس‌دهی می‌توانیم به تمامی عناصر یک تنسور دسترسی داشته باشیم. یک تصویر سطح خاکستری معادل با تنسور دوبعدی و تصویر رنگی معادل با تنسور سه بعدی است. البته، در یادگیری عمیق تنسور چهاربعدی هم خواهید دید! پیشنهاد می‌کنم بخش آشنایی با تنسور در ریاضیات را مطالعه کنید.

بسیارخب، در ادامه توضیح لایه‌های اصلی شبکه عصبی کانولوشن را شروع می‌کنم. اولین لایه، لایه کانولوشنی است. پس برویم با کانولوشن آشنا شویم و بعد هم لایه کانولوشنی را زیر ذره‌بین قرار دهیم...

۴- کانولوشن چیست؟

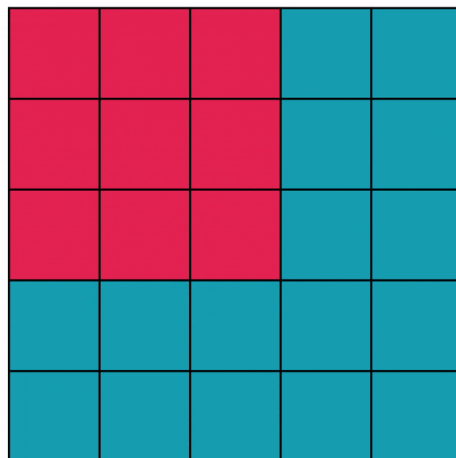
در عملگر کانولوشن، مطابق شکل ۱۰ چهار مولفه مهم وجود دارد که عبارتند از:

- ماتریس یا تصویر ورودی (Input)
- فیلتر یا کرنل کانولوشنی (Convolution Filter)
- عملگر کانولوشن (*)
- ویژگی خروجی کانولوشن (Output)



شکل ۱۰: چهار مولفه اصلی در کانولوشن.

خیلی ساده بخواهم عملکرد کانولوشن را توضیح دهم، باید بگویم که عملگر کانولوشن (*)، کرنل یا فیلتر کانولوشنی را برمی‌دارد و روی تصویر یا ماتریس ورودی می‌لغزاند. به عبارتی دیگر، کرنل یا فیلتر روی تصویر حرکت می‌کند یا تصویر ورودی را اسکن می‌کند. به شکل زیر نگاه کنید؛ ماتریس آبی معادل با تصویر یا ویژگی ورودی و ماتریس قرمز معادل با فیلتر یا کرنل کانولوشنی است. به نحوه حرکت یا اسکن فیلتر روی تصویر دقت کنید. فیلتر ابتدا هر سطر را ستون به ستون طی می‌کند و بعد یک سطر پایین می‌آید و دوباره ستون به ستون جلو می‌رود و این فرآیند تا آخر ادامه دارد...



شکل ۱۱: اسکن شدن ماتریس ورودی با فیلتر کانولوشنی. **انیمیشن**

توجه: در این آموزش یک عالمه شکل با انیمیشن داریم. در کپشن شکل‌هایی که انیمیشن دارند، نوشته‌ایم **انیمیشن**. این انیمیشن‌ها را می‌توانید از لینک [آموزش شبکه عصبی کانولوشن](#) مشاهده کنید. یا اینکه QR Code بالا را اسکن کنید و انیمیشن را ببینید.

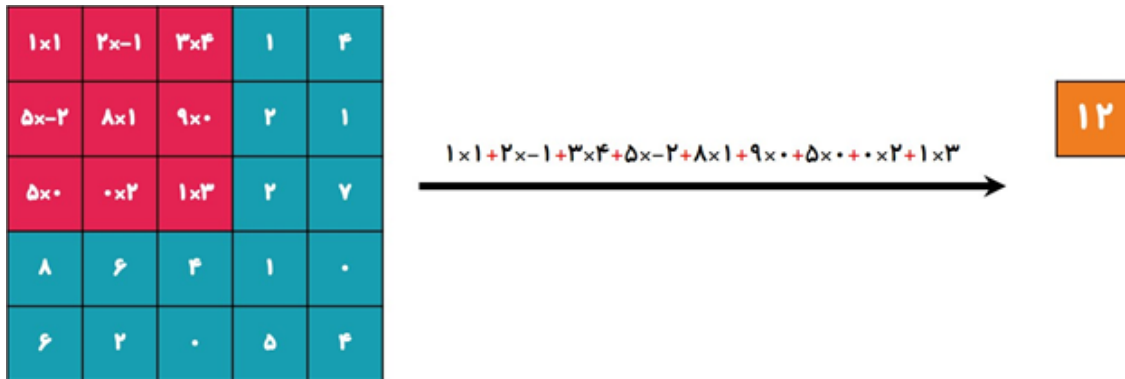
اما ماتریس ورودی و فیلتر بالا شامل هیچ‌گونه عددی نیستند. حالا به تصویر زیر نگاه کنید؛ هر فیلتر کانولوشنی، شامل مجموعه‌ای عدد است. با قرار گرفتن فیلتر روی هر بخش از تصویر، اعداد در فیلتر درایه به درایه در پیکسل‌های متناظر تصویر ضرب می‌شوند. در نهایت همه اعداد باهم جمع می‌شوند. مثلاً در تصویر زیر، ۹ ضرب نظیر به نظیر بین یک بخش 3×3 از ماتریس آبی با ماتریس قرمز انجام شده است. در نهایت، این ۹ ضرب باید باهم به شکل زیر جمع شوند:

$$1 \times 1 + 2 \times -1 + 3 \times 4 + 5 \times -2 + 8 \times 1 + 9 \times 0 + 5 \times 0 + 0 \times 2 + 1 \times 3 = 12$$

1×1	2×-1	3×4	۱	۴
5×-2	8×1	9×0	۲	۱
5×0	0×2	1×3	۲	۷
۸	۶	۴	۱	۰
۶	۲	۰	۵	۳

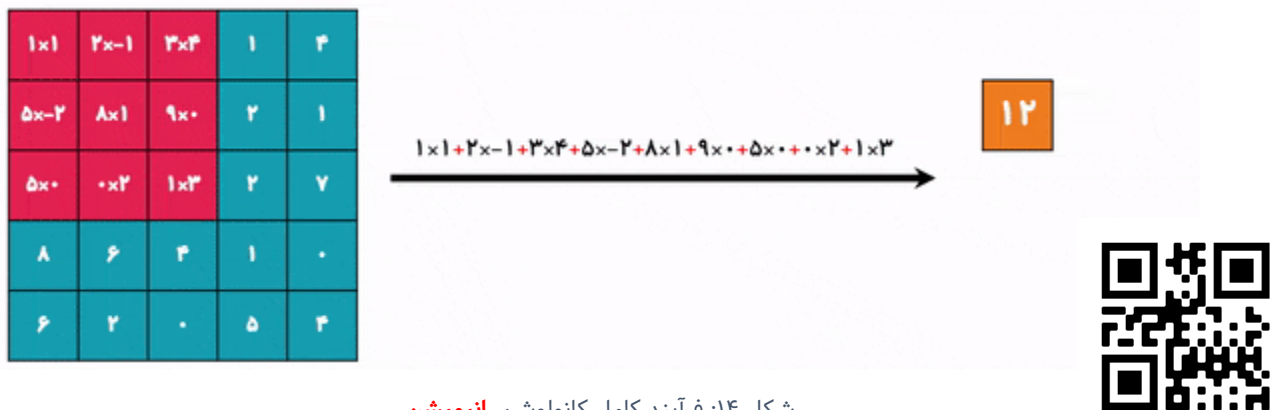
شکل ۱۲: یک مرحله از ضرب بین فیلتر و یک پنجره از ورودی در کانولوشن.

حالا این عدد ۲۵ را کجا باید ذخیره کرد؟ فعلا مطابق شکل زیر، خروجی را در یک آرایه تنها قرار می‌دهیم.



شکل ۱۳: محاسبه خروجی یک مرحله از کانولوشن.

حالا باید فرآیند اسکن کردن (شکل ۱۱) و محاسبه خروجی (شکل ۱۳) باهم ترکیب شود. یعنی فیلتر کانولوشنی سفرهای استانی را شروع کند و در هر محلی خروجی را از طریق ضرب درایه به درایه بین فیلتر و تصویر محاسبه نماید. ضرب‌ها را باهم جمع و در خروجی می‌چینیم. شکل زیر گویای فرآیند کامل کانولوشن است. دقت کنید، هرطوری که فیلتر حرکت می‌کند (مثلا یک ستون جلو می‌رود یا یک سطر پایین می‌آید)، ما هم خروجی را به همان شکل کنار هم می‌چینیم.



شکل ۱۴: فرآیند کامل کانولوشن. **انیمیشن**

۴-۱- چرا سایز خروجی کانولوشن کوچک شد؟!

چرا ماتریس خروجی در شکل ۱۴، دو سطر و ستون کمتر از ماتریس ورودی دارد؟ چرا ماتریس خروجی در شکل ۱۴، ۳×۳ است اما ماتریس ورودی ۵×۵ است؟ چرا هم‌اندازه نیستند؟ جواب این سوال‌ها این است که بسته به اندازه فیلتر، اندازه ماتریس خروجی تغییر می‌کند. مثلا وقتی فیلتر ۳×۳ باشد، خروجی ۲ سطر و ستون کمتر از ورودی خواهد داشت. حالا بیایید این مساله را فرموله کنیم. فرض کنید ماتریس ورودی و فیلتر

مربعی باشند و سایز آنها به ترتیب برابر با k و n باشد. در این صورت، سایز خروجی m با رابطه زیر بدست می آید:

$$m = n - (k - 1)$$

به همین سادگی... حالا اگر سایز ورودی و فیلتر به ترتیب 5×5 و 3×3 باشد، اندازه خروجی برابر با 3×3 خواهد بود. آیا راهی وجود دارد که اندازه ماتریس خروجی کاهش پیدا نکند و همان 5×5 بماند؟ بله وجود دارد، تکنیکی به نام padding یا لایه گذاری ماتریس وجود دارد که به دور ماتریس ورودی سطر و ستون هایی اضافه می شود که باعث می شود جلوی کاهش بعد خروجی گرفته شود. مثلاً، در شکل ۱۴، اگر ورودی را به نوعی جعلی به یک ماتریس 7×7 تبدیل کنیم، آنوقت خروجی 5×5 خواهد بود. یعنی اندازه خروجی برابر با همان ورودی است. دقت کنید؛ ما به صورت جعلی ورودی را افزایش بعد می دهیم. در بخش لایه کانولوشنی راه حلش را توضیح می دهیم.

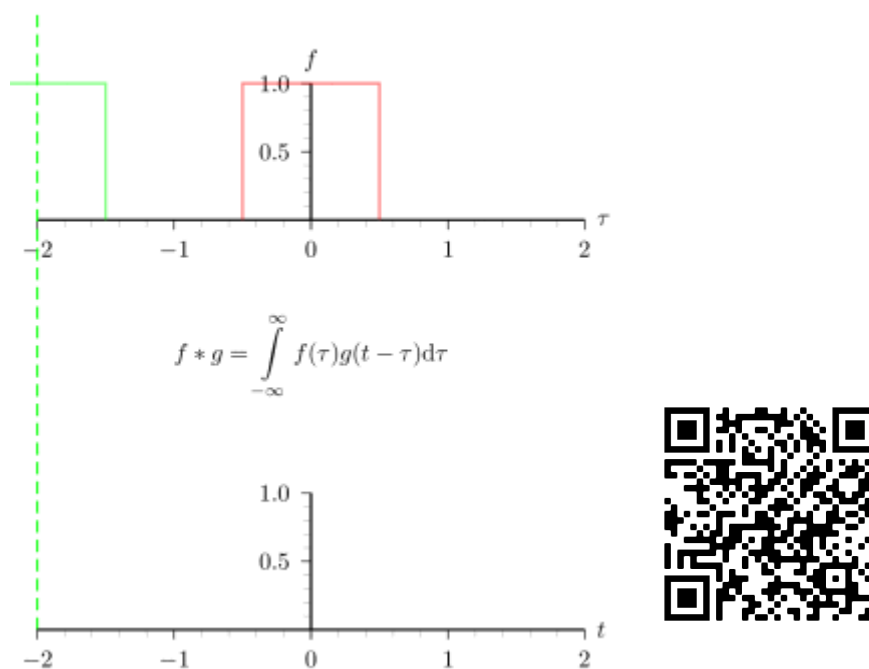
قصد بررسی عمیق عملگر کانولوشن را ندارم، چون خود یک پست جداگانه و طولانی را می طلبد. اما می خواهم خیلی خلاصه درباره کارکرد کانولوشن توضیح دهم. پس بخش بعد را از دست ندهید...

۴-۲- کانولوشن زیر ذره بین

بیا ببینیم چند نکته از کانولوشن را باهم مرور کنیم.

نکته اول: اعدادی که در ماتریس خروجی ذخیره می شوند، تابعی از ورودی و فیلتر هستند. چه زمانی به ازای ضرب بین فیلتر روی یک محل از تصویر، خروجی بزرگ یا کوچک می شود؟ ساده است، هر وقت فیلتر با یک پنجره از تصویر خیلی شبیه هم باشند (از لحاظ عددی)، خروجی عدد بزرگی می شود. اگر هم شبیه هم نباشند، خروجی عدد کوچکی می شود. یعنی چه؟ یعنی اینکه، فیلتر به دنبال پیدا کردن نواحی مشابه خود در تصویر است و هر جایی ناحیه مشابه خود را پیدا کرد بلند فریاد می زند (عدد بزرگ).

پس کانولوشن منجر به یافتن الگوهای خاص در تصویر با توجه به فیلتر می شود. اعداد موجود در فیلتر بسیار مهم هستند. به شکل ۱۵ که یک سیگنال یک بعدی است دقت کنید. ببینید، چطور نواحی مشابه با فیلتر در خروجی آشکار شده است. دقیقاً آنجایی که دو مربع روی هم قرار می گیرند، یک مقدار پیک در خروجی به وجود می آید. یعنی شباهت در آن نقطه بسیار بالاست.



شکل ۱۵: نمونه‌ای از یافتن الگوی مشابه در کانولوشن. **انیمیشن**

نکته دوم، ارتباط دادن عمل کانولوشن به عملکرد یک نورون است. یک نورون یک جمع وزن‌دار (ضرب بین پارامترها و ورودی و نهایتاً جمع) بود. اینجا هم همین است! فیلتر که شامل یک سری اعداد است به ورودی‌ها وزن می‌دهد (ضرب درایه به درایه بین فیلتر و پنجره‌های ماتریس ورودی) و نهایتاً اعداد وزن‌دهی‌شده ورودی را باهم جمع می‌کند.

نکته سوم: در نورون پارامترها متغیر بودند و از طریق فرآیند آموزش بدست می‌آمدند. اینجا هم اعداد موجود در فیلتر از طریق فرآیند آموزش به دست می‌آیند. البته، قبل از پیدایش شبکه عصبی کانولوشن، از کانولوشن در پردازش تصویر و سیگنال بسیار زیاد استفاده می‌شد. اما اعداد فیلترهای کانولوشنی ثابت بودند و توسط یک متخصص این اعداد طراحی می‌شوند. مثلاً در شکل ۱۶ یک فیلتر لبه‌یابی با نتیجه خروجی را آورده‌ام. ببینید، فیلترها چه بلایی سر تصویر می‌آورند و فقط دنبال مطلوب خود هستند.



$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

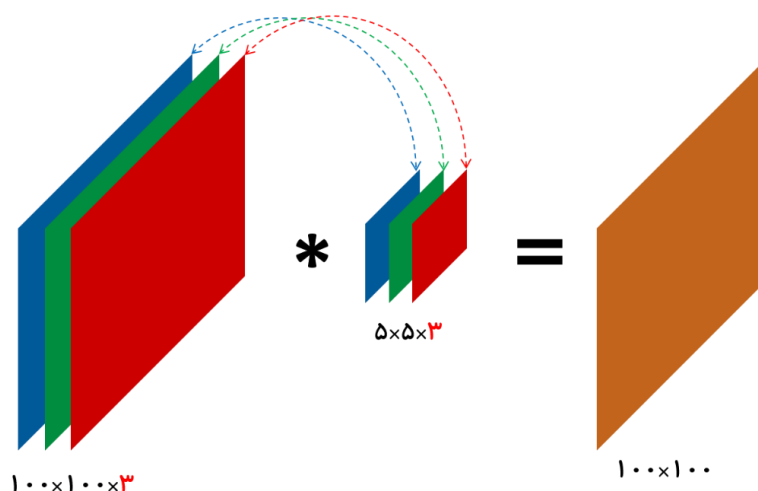


شکل ۱۶: پیدا کردن لبه‌های تصویر ورودی با استفاده از کانولوشن.

تا اینجا، به شما درباره کانولوشن تصویر دوبعدی (تصویر سطح خاکستری) توضیح دادم. اما در شبکه عصبی کانولوشن، معمولاً ورودی یک تصویر رنگی (ماتریس سه بعدی) است. در این حالت کانولوشن چگونه انجام می‌شود؟ بسیار ساده است...

۳-۴- کانولوشن با ورودی سه‌بعدی

همان‌طور که در بخش قبلی گفتم، تصویر رنگی سه کانال یا صفحه دارد. بنابراین، فیلتر شما هم باید سه کانال داشته باشد. یعنی اگر قبلاً یک فیلتر 3×3 برای کانولوشن در تصویر دوبعدی داشتیم، حالا یک فیلتر $3 \times 3 \times 3$ خواهیم داشت. چرا؟ چون هر صفحه از تصویر رنگی به صورت مجزا برای خودش یک فیلتر دارد. به شکل ۱۷ نگاه کنید؛ سه صفحه از فیلتر به صورت موازی باهم روی سه صفحه از تصویر حرکت می‌کنند. با فرض اینکه فیلتر 5×5 باشد، در هر لحظه ۲۵ ضرب در هر صفحه انجام می‌شود و در مجموع برای سه صفحه می‌شود ۷۵ ضرب... در نهایت خروجی برابر است با جمع این ۷۵ ضرب. پس در نهایت نتیجه ضرب سه صفحه باهم جمع می‌شود و خروجی نهایی یک تصویر $3 \times 100 \times 100$ نیست، بلکه یک تصویر 100×100 است.



شکل ۱۷: کانولوشن در تصویر رنگی.

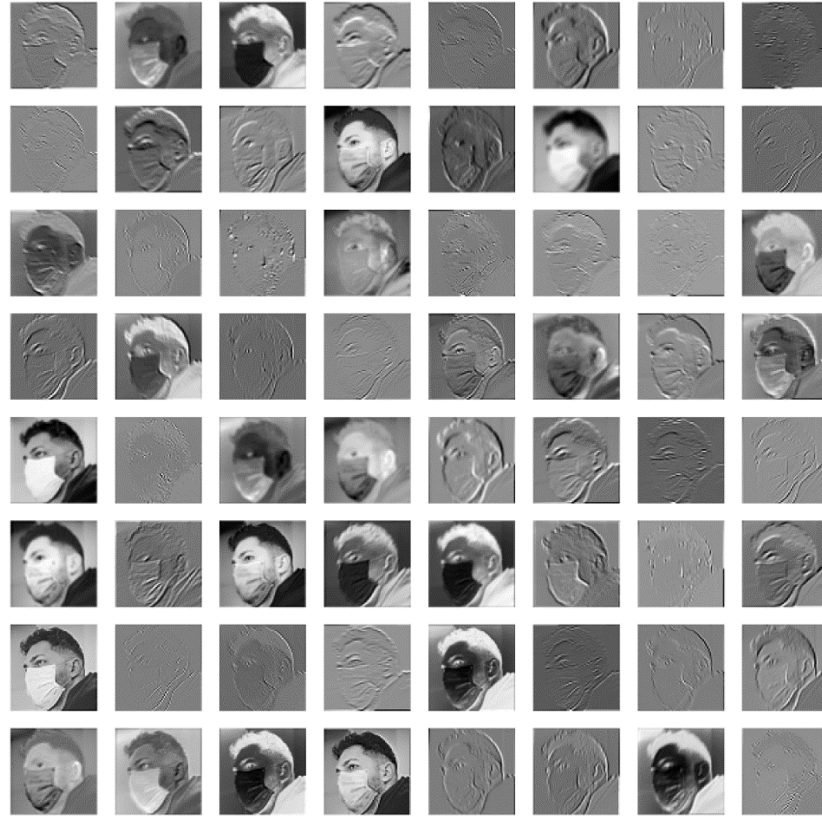
حالا مثلا مثلا مثلا، اگر ورودی بجای ۳ صفحه، ۱۰ صفحه داشته باشد، برای خروجی چه اتفاقی می افتد؟ فیلتر چه شکلی خواهد بود؟ فیلتر باید ۱۰ صفحه داشته باشد و اگر فرضا فیلتر را 5×5 در نظر بگیریم، اندازه فیلتر به شکل $5 \times 5 \times 10$ باشد. هریک از این ۱۰ صفحه، در صفحه متناظر با صفحه تصویر کانوالو می شود. بعد از توضیح درباره کانولوشن، وقت آن رسیده که درباره لایه کانولوشنی در شبکه عصبی CNN توضیح بدهم. برویم سراغ بخش بعدی...

۵- لایه کانولوشن در شبکه عصبی کانولوشن (Convolutional Layer)

هسته اصلی شبکه CNN لایه کانولوشنی است که درصد اعظم محاسبات شبکه عصبی کانولوشن را به خود اختصاص داده است. هر لایه کانولوشن در شبکه عصبی کانولوشن شامل مجموعه ای فیلتر است و از کانولوشن بین فیلترها و لایه ورودی است که خروجی ساخته می شود. به خروجی لایه کانولوشنی، فیچر مپ (Feature Map) گفته می شود. می خواهم این جمله را کمی بشکافم.

همان طور که بخش قبلی گفتم، یک فیلتر می تواند شامل یک الگویی خاص باشد و در تصویر به دنبال آن الگو باشد. اتفاقا در فرآیند آموزش شبکه، به دنبال این هستیم که این فیلترها الگوهای معناداری از هر تصویر استخراج کنند. مثلا، فیلتری داریم که شامل الگوی گوش گربه هست و می تواند حضور یک گربه در تصویر ورودی را تشخیص دهد. اما فقط به یک الگو بسنده کنیم؟ آیا گربه، فقط شامل یک الگوی خاص است؟ مثلا، دهان، گوش ها، چشم ها و دم هر کدام الگوی خاصی از گربه نیستند؟ با یک فیلتر می شود همه این الگوها را شکار کرد؟ بهتر نیست شواهد بیشتری جمع کنیم و بعد تصمیم بگیریم؟

این همان کاری است که در لایه کانولوشنی انجام می شود. جستجو در تصویر برای یافتن تنها یک الگو منجر به نتایج خوبی نمی شود و باعث می شود شبکه از لحاظ کارایی محدود باشد. برای حل این مشکل، نیاز است که لایه کانولوشنی چندین فیلتر داشته باشد. هریک از فیلترها به تنهایی یک الگوی خاص داشته باشند و خروجی لایه کانولوشنی مجموعه ای از الگوهای مختلف باشد. در شکل زیر، تصویری نشان داده شده که یک تصویر با فیلترهای مختلفی کانوالو شده و هر فیلتر هم یک خروجی جداگانه برای خود تولید کرده است. همه تصاویر مربوط به یک چهره با ماسک هست؛ اما دقت کنید که در هر تصویری یک مجموعه ویژگی خاص مورد توجه بوده است. مثلا در تصویری، لبه های افقی، در دیگری لبه های عمودی و غیره. در شکل زیر ۶۴ فیلتر مختلف به تصویر ورودی اعمال شده است.



شکل ۱۸: اعمال ۶۴ فیلتر مختلف به یک تصویر با لایه کانولوشنی.

اما، یک لایه کانولوشنی در شبکه عصبی کانولوشن را چگونه نمایش می‌دهند؟ بریم ببینیم...

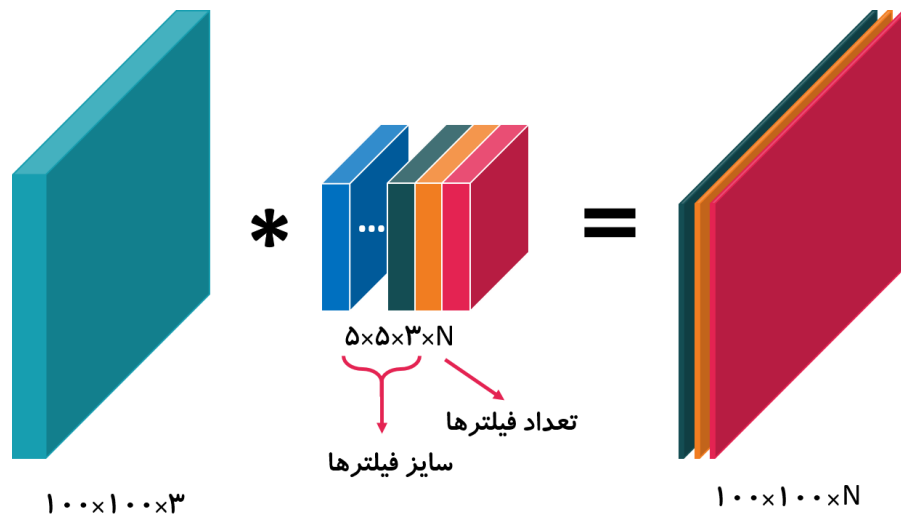
۵-۱- نمایش لایه کانولوشن

نمایش بلوک دیاگرامی شکل زیر، یک نمایش مرسوم برای لایه کانولوشنی است. حالا جزیه جز درمورد شکل ۱۹ توضیح می‌دهم؛ اول، تکلیف ورودی را مشخص کنم که یک تانسور سه بعدی است. دوم، فیلترها را مشاهده می‌کنید. دیگر یک فیلتر نداریم بلکه یک مجموعه فیلتر داریم (در اینجا N تا فیلتر). معمولاً همه فیلترها را یک‌جا در قالب یک تانسور چهاربعدی نمایش می‌دهند. از سمت چپ به راست، هریک از ابعاد موجود در فیلتر معادل با موارد زیر هستند:

- بعد اول، معادل با تعداد سطر (۵)
- بعد دوم، معادل با تعداد ستون (۵)
- بعد سوم، معادل با تعداد کانال (۳)
- بعد چهارم، معادل با تعداد فیلتر (N)

یعنی یک تانسور چهاربعدی به ابعاد $5 \times 5 \times 3 \times N$ معادل با N فیلتر به ابعاد $5 \times 5 \times 3$ است. پس قرار است روی تصویر ورودی، N فیلتر اعمال شود. شما به عنوان یک متخصص مجاز هستید که اندازه سطر و ستون و تعداد فیلترها را خودتان تعیین کنید. اما بعد سوم (تعداد کانالها)، از روی تعداد کانال ورودی تعیین می‌شود. چرا؟ خودتان دلیلش را بگویید، در آخر بخش قبلی توضیح دادم.

درنهایت، خروجی یک فیچرماپ است به ابعاد $100 \times 100 \times N$ که عدد سوم برابر با تعداد فیلترهاست. چرا؟ چون هر فیلتر یک خروجی تولید می‌کند و با N فیلتر، ما N صفحه خواهیم داشت. بنابراین، مطابق شکل زیر خروجی یک لایه کانولوشنی یک تانسور سه بعدی خواهد بود. اگر به شکل ۱۸ دوباره نگاه کنید، می‌بینید که تعداد فیلترها برابر با $N=64$ است و خروجی هم یک تانسور سه بعدی با 64 صفحه هست. البته، در شکل ۱۸، این 64 صفحه باز شده و به صورت جداگانه نمایش داده شده است.



شکل ۱۹: بلوک دیاگرام لایه کانولوشنی با N فیلتر.

خب، انتظار دارم با لایه کانولوشنی آشنا شده باشید. اما همواره با کانولوشن، مفاهیم یا پارامترهای مهمی مطرح می‌شوند که دانستن آنها ضروری است. در ادامه، این پارامترها را به شما معرفی می‌کنم.

۵-۲- تعداد فیلتر در لایه کانولوشن

تعداد فیلترها، تعداد ویژگی‌های تشخیص داده شده را نشان می‌دهد. این هایپرپارامتر معمولاً نمایی از ۲ (معمولاً بین ۳۲ تا ۴۰۹۶) انتخاب می‌شود. معمولاً استفاده از فیلترهای بیشتر باعث ایجاد یک شبکه عصبی قدرتمندتر می‌شود. اما این افزایش پارامترها ممکن است باعث overfitting شود.

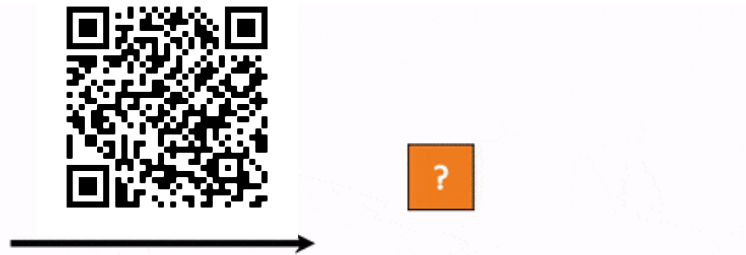
۵-۳- اندازه فیلتر در لایه کانولوشن

اندازه فیلتر معمولاً به صورت مربعی (برای تصاویر) و 3×3 تعریف می‌شود. اما 5×5 و 7×7 نیز گاهی استفاده می‌شود. با فیلترهای کوچک، تعداد پارامترهای قابل یادگیری بسیار کمتر می‌شود.

۵-۴- اندازه پدینگ (padding) در لایه کانولوشن

در بخش قبل دیدید که بعد از فیلتر کردن یک ماتریس 5×5 با فیلتر 3×3 ، سایز خروجی دو سطر و ستون کمتر از ماتریس ورودی داشت. گفتیم که با لایه گذاری یا padding می‌توانیم به صورت جعلی اندازه ورودی را افزایش دهیم تا ماتریس خروجی هم‌اندازه ماتریس ورودی بشود. درمورد لایه گذاری صحبت بسیار است و معمولاً در پردازش تصویر بررسی می‌شود. اما یک راه ساده و رایج آن اضافه کردن سطر و ستون صفر به صورت متقارن به دور ماتریس ورودی است. به لایه گذاری صفر zero padding گفته می‌شود. به شکل ۲۰ نگاه کنید که چگونه یک لایه صفر دور ماتریس قرار گرفته است. حالا فیلتر کانولوشنی ما فضای بیشتر برای گام برداشتن و اسکن کردن دارد و طبیعتاً خروجی هم بزرگتر می‌شود.

$\times 1$	$\times -1$	$\times 3$
.
$\times -2$	$\times 1$	$\times 0$	۳	۱	۴	.
.	۱	۲
$\times 0$	$\times 2$	$\times 3$	۹	۲	۱	.
.	۵	۸
.	۵	.	۱	۲	۷	.
.	۸	۶	۴	۱	.	.
.	۶	۲	.	۵	۴	.
.



شکل ۲۰: کانولوشن با padding. انیمیشن

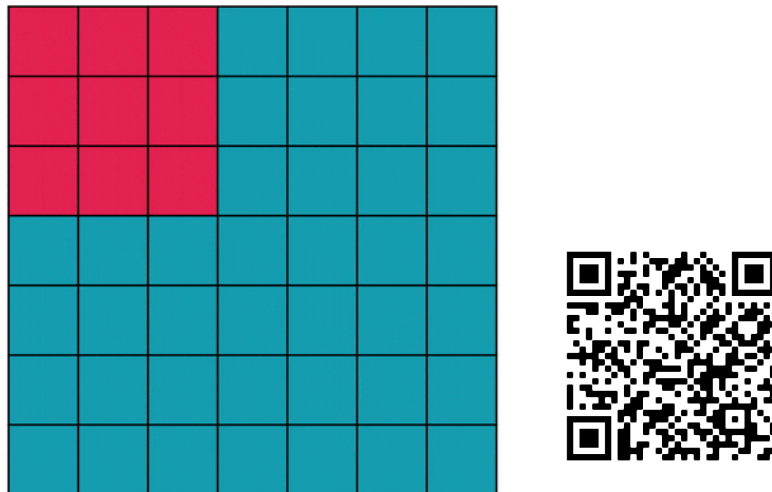
تمرین: علامت‌های سوال شکل بالا را پر کنید!

از کجا بفهمیم چه تعداد لایه باید دور ماتریس اضافه کرد؟ بستگی به اندازه فیلتر دارد؛ قبلاً هم گفته بودم که به اندازه $\frac{n - (k-1)}{2}$ از ماتریس خروجی کاسته می‌شود. بسیار خب، واضح است که به همین اندازه باید سطر و ستون به ماتریس ورودی اضافه کنیم. مثلاً فرمول $\frac{n - (k-1)}{2}$ برای فیلتر 3×3 می‌گوید که باید به ماتریس

ورودی ۲ سطر و ستون اضافه شود. یک سطر را بالای ماتریس ورودی قرار می‌دهیم و یکی را هم پایین ماتریس... یک ستون را سمت چپ ماتریس و دیگری را سمت راست... به عنوان مثالی دیگر، اگر اندازه فیلتر 5×5 باشد، چهار سطر و ستون باید به ماتریس ورودی اضافه شود.

۵-۵- اندازه stride در لایه کانولوشن

اندازه stride را می‌توان گام هم تعریف کرد. منظور از گام این است که فیلتر بعد از محاسبه در یک پنجره از ورودی، چند درایه یا خانه باید جلو برود تا دوباره محاسبات را انجام دهد. اگر اندازه گام را ۱ در نظر بگیریم، یعنی فیلتر باید درایه به درایه در ورودی به سمت راست و پایین برود و اسکن را انجام دهد. یعنی، همان صحبت‌هایی که در بخش قبل (کانولوشن) داشتیم. اما اگر مثلاً stride عدد ۲ در نظر گرفته شود، طبق شکل ۲۱ مشاهده می‌کنید که دوتا دوتا می‌پرد.



شکل ۲۱: تعریف ۲ stride در شبکه عصبی کانولوشن. **انیمیشن**

نکته: معمولاً stride در لایه های کانولوشنی عدد ۱ در نظر گرفته می‌شود.

عملگر کانولوشن، یکی از مهم‌ترین مولفه‌هایی است که باعث می‌شود CNN نسبت به تغییرات مکانی مقاوم باشد. برای یک فیلتر با یک الگوی گربه، فرقی ندارد که گربه در کجای تصویر قرار دارد. در هر صورت آن گربه در تصویر را تشخیص می‌دهد. چون فیلتر کل تصویر را اسکن می‌کند و فارغ از موقعیت، گربه را تشخیص دهد.

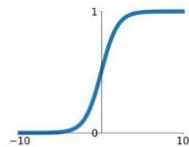
۶- لایه غیرخطی در شبکه عصبی کانولوشن (Activation Function)

مشابه با سایر شبکه های عصبی، شبکه عصبی کانولوشن هم از تابع تحریک یا activation function غیرخطی بعد از لایه کانولوشنی استفاده می کند. استفاده از تابع غیرخطی باعث ایجاد خاصیت غیرخطی در شبکه عصبی می شود که خیلی مهم است. بعضی از فریمورک ها در همان تعریف لایه کانولوشنی، به شما این امکان را می دهند که نوع تابع غیرخطی را هم مشخص کنید. در بعضی از فریمورک ها هم باید یک لایه جداگانه بسازید. تعریف تابع غیرخطی به صورت جدا از لایه کانولوشنی انعطاف پذیری بیشتری ایجاد می کند.

در بین تمام توابع غیرخطی، تابع ReLU بیشترین محبوبیت را دارد. البته، از خانواده ReLU اعضای دیگری مانند PReLU Leaky-ReLU و غیره وجود دارند. در شکل زیر، تعدادی از توابع غیرخطی را مشاهده می کنید. به عنوان نمونه، تابع ReLU مقادیر کوچکتر از صفر (منفی) را صفر و مقادیر بزرگتر از صفر را بدون هیچ گونه تغییری به خروجی می برد.

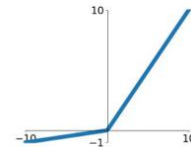
Sigmoid

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



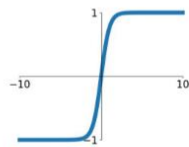
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

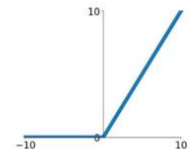


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

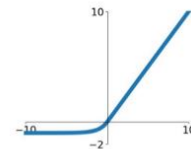
ReLU

$$\max(0, x)$$



ELU

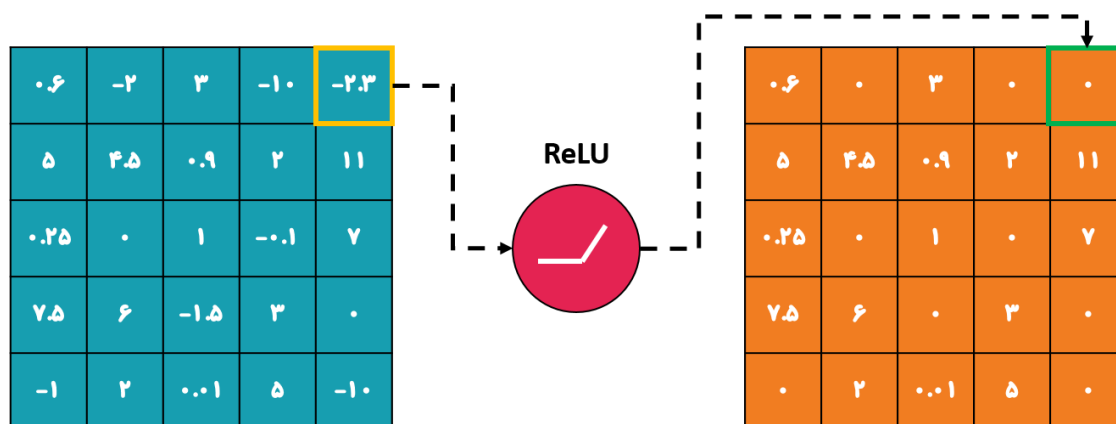
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



شکل ۲۲: توابع غیرخطی در شبکه عصبی کانولوشن.

محاسبات موجود در تابع ReLU ساده و صرفاً یک مقایسه است. به همین خاطر محاسبات در بخش تابع غیرخطی با استفاده از ReLU نسبت به سایر توابع غیرخطی (مانند sigmoid tanh) با سرعت بیشتری انجام می شود. از طرفی فرآیند آموزش با ReLU نسبت به سایر توابع غیرخطی سریع تر است. چون، توابع غیرخطی tanh sigmoid در مقادیر خیلی بزرگ و کوچک به اشباع می رسند و این باعث می شود که گرادیان این توابع به سمت صفر میل کند. در نتیجه کل فرآیند بهینه سازی یا بهتر بگوییم آموزش با سرعت پایین تری نسبت به

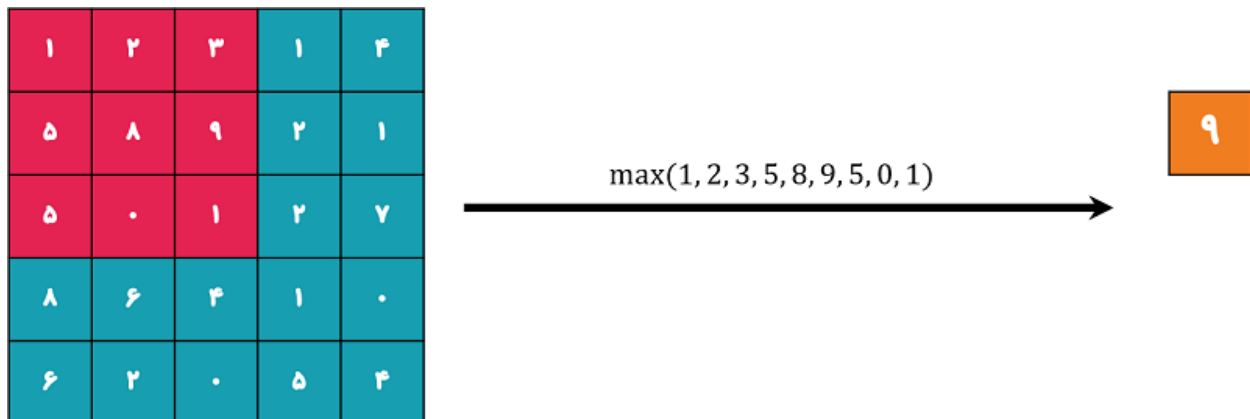
ReLU انجام می‌شود. در شکل زیر، خروجی یک نمونه اعمال تابع تحریک ReLU به یک ماتریس ورودی را مشاهده می‌کنید.



شکل ۲۳: تابع تحریک ReLU و نحوه اعمال به یک ورودی نمونه.

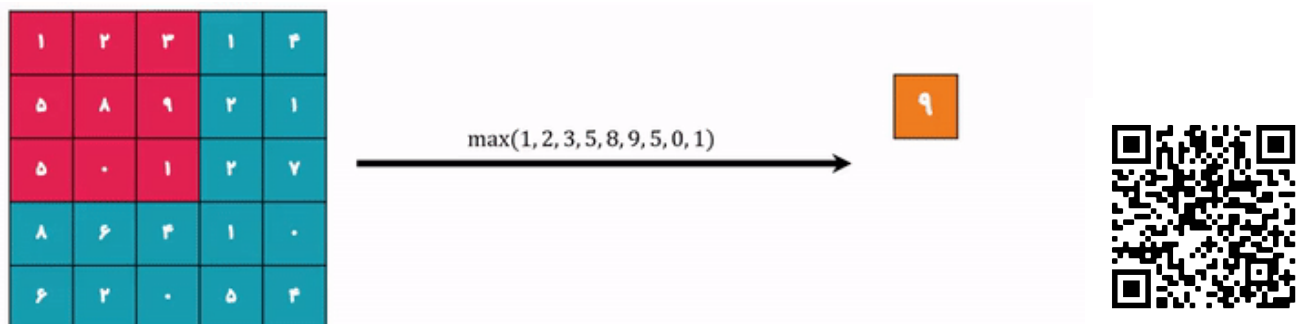
۷- لایه پولینگ در شبکه عصبی کانولوشن (Pooling Layer)

لایه پولینگ یکی دیگر از لایه های مهم در شبکه عصبی کانولوشن است. هدف لایه پولینگ کاهش اندازه مکانی فیچرکمپ بدست آمده با استفاده از لایه کانولوشنی است. لایه پولینگ پارامتر قابل آموزش ندارد. صرفاً یک نمونه برداری ساده و موثر انجام می‌دهد. پولینگ عملکردی شبیه کانولوشن دارد و یک پنجره روی تصویر حرکت می‌کند. رایج‌ترین نمونه پولینگ max pooling و average pooling است. مطابق شکل ۲۴ ماکس پولینگ شامل یک پنجره از پیش تعریف شده است (مثلاً 3×3) که روی تصویر حرکت می‌کند (تصویر را اسکن می‌کند) و در هر پنجره مقدار ماکزیمم را انتخاب می‌کند و بقیه را دور می‌ریزد. مشابه با لایه کانولوشنی، اندازه فیلتر، پدینگ و استراید در اینجا هم صدق می‌کند. در پولینگ، معمولاً استراید ۲ و اندازه فیلتر ۳ در نظر گرفته می‌شود. برای اینکه سایز ویژگی به $2/1$ کاهش پیدا کند.



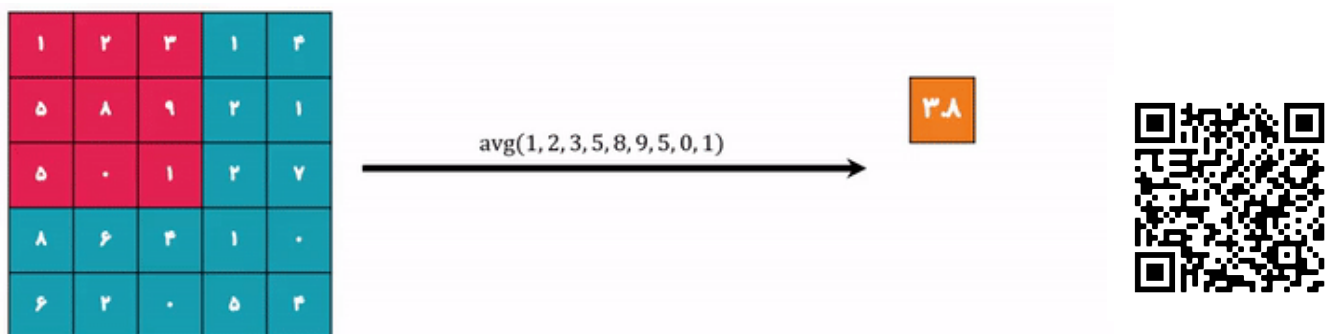
شکل ۲۴: نمایی از نحوه عملکرد max pooling در شبکه عصبی کانولوشن.

در شکل زیر یک نمونه کامل از max pooling نشان داده شده است.



شکل ۲۵: عملکرد max pooling در شبکه عصبی کانولوشن. **انیمیشن**

همچنین، در شکل زیر یک نمونه کامل از average pooling نشان داده شده است.



شکل ۲۶: عملکرد average pooling در شبکه کانولوشنی. **انیمیشن**

صبر کنید، صبر کنید! تا اینجا لایه کانولوشن، لایه غیرخطی و لایه پولینگ را آموختید. اما قبل از اینکه بخش‌های بعدی را توضیح بدهم، می‌خواهم یک شبکه عصبی کانولوشن نصف و نیمه با همین مازول‌ها بسازم.

برای ساخت یک شبکه کانولوشنی ساده لازم است نکاتی را مدنظر داشته باشید. بخش بعدی را دقیق بخوانید...

۸- طراحی شبکه عصبی کانولوشن ساده

من در شکل ۲۷ یک شبکه کانولوشن نصف و نیمه ساختم. به شکل نگاه کنید و نکته‌های زیر را بخوانید:

۱- معمولا در شبکه عصبی کانولوشن، ابتدا لایه کانولوشنی قرار می‌گیرد. یعنی، قبل از پولینگ، تابع غیرخطی و غیره. ابتدا، یک لایه کانولوشنی روی تصویر با تعداد فیلتر کم اعمال می‌شود. لایه ۱، با $32 \times 32 \times 32$ فیلتر را در شکل زیر مشاهده می‌کنید. پدینگ و استراید طوری تنظیم شده که سایز فیچرماپ خروجی کم نشود. اندازه فیچرماپ خروجی برابر با $32 \times 32 \times 32$ است.

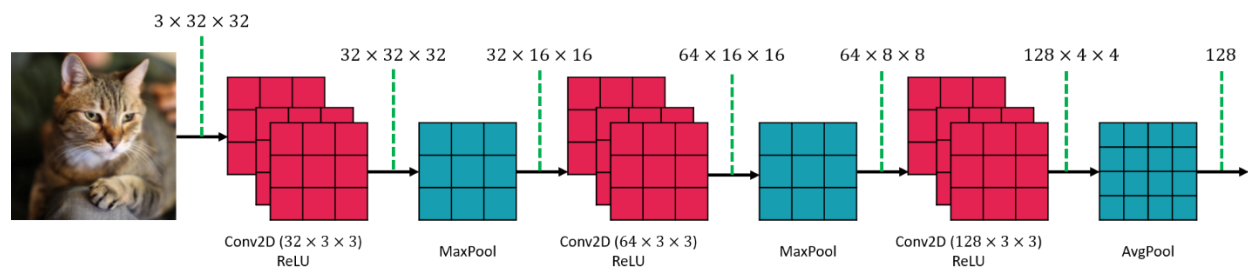
۲- معمولا بعد از لایه کانولوشنی، یک تابع تحریک قرار داده می‌شود. طبیعی است، چون اگر یادتان باشد در نورون هم بعد از جمع وزن‌دهی و جمع با بایاس، تابع تحریک اعمال می‌شد. طبق شکل زیر، یک تابع ReLU روی فیچرماپ اعمال می‌شود. کارش هم اینست که منفی‌ها را صفر می‌کند، تمام!

۳- حالا می‌خواهم یک لایه پولینگ بعد از تابع تحریک قرار دهم. این پولینگ از نوع ماکس پولینگ است. اما استراید را ۲ در نظر گرفتم که اندازه فیچرماپ خروجی نصف شود. پس اندازه فیچرماپ خروجی برابر با $32 \times 16 \times 16$ خواهد شد.

۴- تا اینجا یک دور کانولوشنی، غیرخطی و پولینگ را به تصویر ورودی اعمال کردم. دوباره هر سه اینها را اعمال می‌کنم. تعداد فیلترهای لایه کانولوشنی دوم را ۶۴ در نظر گرفتم. فیچرماپ خروجی ماکس پولینگ: $64 \times 8 \times 8$

۵- در سومین مرحله، یک لایه کانولوشنی با ۱۲۸ فیلتر در نظر گرفته شده و استراید هم ۲ است. بعد از لایه کانولوشنی یک تابع تحریک گذاشتم. فیچرماپ خروجی تابع تحریک برابر با $128 \times 4 \times 4$ است.

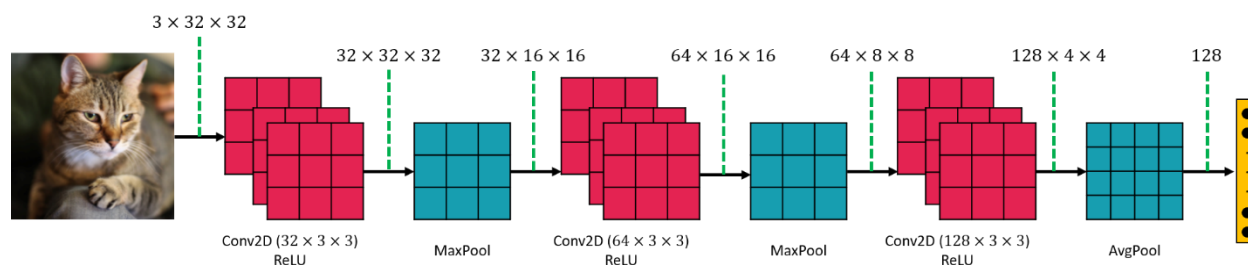
۶- در آخر یک لایه میانگین پولینگ با سایز 4×4 گذاشتم. فیچرماپ خروجی این لایه برابر با $128 \times 1 \times 1$ خواهد شد. یعنی، تصویر ورودی بعد از چندمرحله تبدیل به یک بردار به طول ۱۲۸ شد. این بردار، بردار ویژگی تصویر ورودی است. این بردار شناسنامه یا اثر انگشت تصویر است. به چه دردی می‌خورد؟ در بخش بعدی خواهید دید...



شکل ۲۷: طراحی شبکه عصبی کانولوشن ساده.

۹- لایه فولی کانکتد در شبکه کانولوشن (Fully Connected Layer)

معمولاً آخرین لایه های یک شبکه عصبی کانولوشن برای طبقه بندی را لایه های فولی کانکتد تشکیل می دهند. این لایه ها همان لایه هایی هستند که در شبکه عصبی MLP دیده اید. برای آشنایی بیشتر این مقاله را بخوانید. یکی از کاربردهای اصلی لایه فولی کانکتد در شبکه کانولوشن، استفاده به عنوان طبقه بند یا کلاسیفایر (Classifier) است. یعنی مجموعه ویژگی های استخراج شده با استفاده از لایه های کانولوشنی در نهایت تبدیل به یک بردار می شوند. در نهایت این بردار ویژگی به یک کلاسیفایر فولی کانکتد داده می شود تا کلاس درست را شناسایی کند. شکل ۲۸ را با فولی کانکتد کامل کردم و می توانید در شکل زیر مشاهده کنید. شکل زیر بلوک دیاگرام کامل یک شبکه عصبی کانولوشن ساده برای طبقه بندی است.



شکل ۲۸: شبکه عصبی کانولوشن برای طبقه بندی.

۱۰- سوالات رایج درباره شبکه عصبی کانولوشن

انتظار دارم بعد از خواندن این پست بلن دوبالا، یک عالمه سوال برایتان ایجاد شده باشد. من به تعدادی از سوالات رایج دانشجویان در ادامه جواب داده ام. سعی می کنم این بخش را بازم با سوالات جدید تکمیل کنم. سوالاتی مثل:

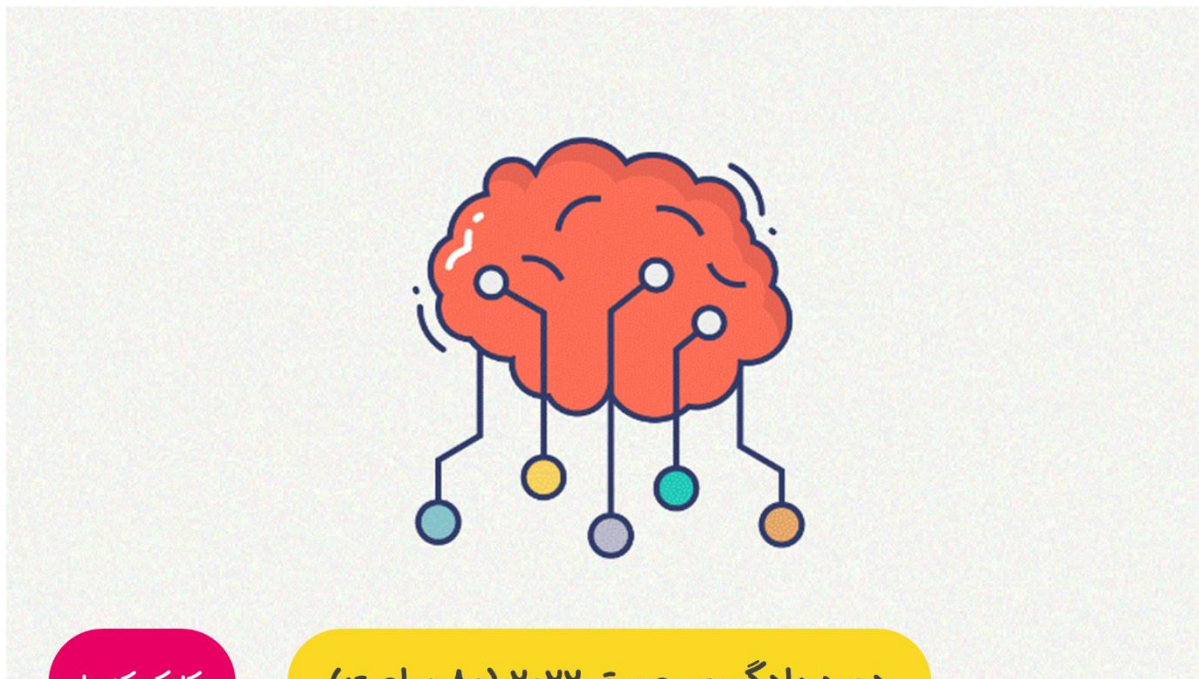
- همیشه اول لایه کانولوشنی، تابع تحریک و بعد هم پولینگ؟ خیر، ممکن است تصمیم بگیریم چندلایه پشت هم کانولوشنی بگذاریم و بعد یک پولینگ قرار دهیم. برچه اساسی این کار انجام می‌شود؟ جواب این سوال سخت است، باید مقاله‌های مختلف در حوزه شبکه عصبی کانولوشن را بخوانید.
- چرا رفته رفته در شبکه جلوتر رفتیم، تعداد فیلترهای لایه کانولوشن بیشتر شد؟ سعی می‌کنیم رفته رفته ویژگی‌های بیشتر و متنوع‌تری تولید کنیم.
- ماکس پولینگ و پولینگ میانگین را برچه اساسی انتخاب می‌کنیم؟ معمولا در لایه‌های میانی از ماکس پولینگ استفاده می‌شود و در پایان شبکه لایه پولینگ میانگین به کار گرفته می‌شود.
- حالا این بردار ویژگی ۱۲۸-تایی به چه دردی می‌خورد؟ همان ابتدا گفتم که من می‌خواهم یک شبکه عصبی کانولوشن برای طبقه بندی بسازم. این خروجی به من کمک می‌کند که من فرق بین گربه و سگ را شناسایی کنم. باید بردارهای ویژگی ۱۲۸-تایی به گونه‌ای باشد که بردار ویژگی تمام تصاویر گربه شبیه هم باشد. همچنین، بردار ویژگی تصاویر سگ مشابه هم باشد، اما بردار ویژگی سگ و گربه باید از هم دور باشند. دو مجموعه بردار ویژگی داریم که گربه‌ها نزدیک هم و سگ‌ها نیز نزدیک به هم است. پس، بردار ویژگی یک تصویر جدید تستی، به هرکدام از این دو بردار که بیشتر شبیه باشد، به همان کلاس متعلق هست.

۱۱- واقعا به پایان رسیدم؟!

جواب کوتاه اینکه، نه! 😊 اما، خدایوت، گام بزرگی برداشتید. از ابتدا تا انتهای یک شبکه CNN را خدمت شما توضیح دادم. امیدوارم، با این شبکه محبوب ارتباط برقرار کرده باشید. مفاهیم و لایه‌های دیگری نیز در شبکه عصبی کانولوشن وجود دارد. مثلا، باید درباره Batch Normalization, Dropout, Data Augmentation, Receptive Field هم توضیح بدهم. لطفا فیدبک بدهید، این آموزش چطور بود؟ در کامنت درباره این پست با ما صحبت کنید. هوسم را هم فراموش نکنید و لطفا به دوستانتان هم معرفی کنید...

<https://howsam.org/convolutional-neural-network/>

دوره‌های مرتبط با شبکه عصبی کانولوشن



کلیک کنید!

دوره یادگیری عمیق ۲۰۲۲ (۸۰ ساعت)

در این دوره مطالب پیشرفته‌تری برای شما داریم!

خصوصیات مهم دوره یادگیری عمیق ۲۰۲۲:

- سرفصل‌های به‌روز و جذاب (شامل مبحث ترنسفورمر، GNN و ...)
- پشتیبانی تلگرامی دوره
- انیمیشن‌های جذاب تالیفی برای فهم بهتر مفاهیم
- کدنویسی پایتورچ



۴۰ درصد تخفیف برای شما بابت همراهی



کد تخفیف: **CNN4PDF**

مشاهده نمونه ویدئو و سیلابس