

مقدمه ای بر شبکه های عصبی پیچشی

نوشته *Ryan Nash و Keiron O'Shea*

ترجمه محمد محمدزاده

خلاصه: حوزه یادگیری ماشین در چند وقت اخیر با ظهور شبکه های عصبی مصنوعی (ANN) پیچ و تاب های چشمگیری داشته است. این مدل های محاسباتی الهام گرفته شده از زیست شناسی، قادر هستند از عملکرد شکل های پیشین هوش مصنوعی در کارهای معمولی یادگیری ماشین بسیار فراتر رفته. یکی از تحسین برانگیز ترین شکل های معماری ANN مربوط به شبکه های عصبی پیچشی (CNN) است. CNN ها عمدتاً برای حل کارهای تشخیص الگوی مبتنی بر تصویر دشوار استفاده میشوند و با معماری دقیق و در عین حال ساده شان روشی ساده شده برای شروع کار با شبکه های عصبی مصنوعی ارائه میکنند.

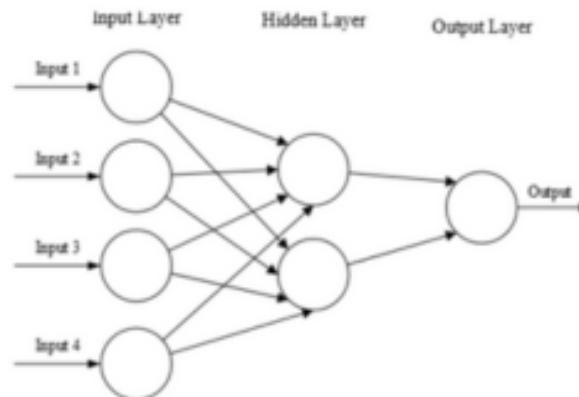
این مقاله، مقدمه خلاصه ای به CNN ها ارائه میکند و درمورد مقالات اخیر و تکنیک های به تازگی شکل گرفته در توسعه این مدل های تشخیص تصویر فوق العاده و هوشمندانه بحث میکند. این مقدمه تصور میکند شما با اصول اولیه ANN ها و یادگیری ماشین آشنا هستید.

کلمات کلیدی: تشخیص الگو، شبکه های عصبی مصنوعی، یادگیری ماشین، آنالیز تصویر

1 مقدمه

شبکه های عصبی مصنوعی (ANN) سیستم های محاسباتی پردازش کننده ای هستند که به شدت از طرز عملکرد سیستم های عصبی زیستی (مثل مغز انسان) الهام گرفته شده اند. ANN ها بطور عمده شامل تعداد زیادی نود های محاسباتی به هم مرتبط هستند (که به آنها نورون گفته میشود) که بصورت در هم پیچیده و توزیع شده در هم کار میکنند تا به طور جمعی به منظور بهینه سازی خروجی، از ورودی یاد بگیرند.

ساختار پایه یک ANN میتواند همانند شکل 1 مدلسازی شود. ما ورودی را که معمولاً به شکل یک بردار چند بعدی است در لایه ورودی بارگذاری میکنیم و این لایه ها ورودی را به لایه های پنهان توزیع خواهند کرد. سپس این لایه های پنهان تصمیماتی را از لایه پیشین میگیرند و ارزیابی میکنند که چطور یک تغییر تصادفی درون خود، خروجی نهایی را بهبود میبخشد یا تضعیف میکند. به این فرآیند یادگیری میگوییم. داشتن چندین لایه پنهان که روی هم قرار گرفته اند معمولاً یادگیری عمیق نامیده میشود.



شکل 1: یک شبکه عصبی پیشخور (FNN) ساده سه لایه که از یک لایه ورودی، یک لایه پنهان و یک لایه خروجی تشکیل شده است. این ساختار پایه چندین معماری ANN رایج است، از جمله ماشین بولتزمن محدود (RBM)، شبکه عصبی بازگشتی (RNN) و ...

دو الگوی یادگیری کلیدی در کارهای پردازش تصویر، یادگیری های نظارت شده و بدون نظارت هستند. **یادگیری نظارت شده**، یادگیری از طریق ورودی های از قبل نامگذاری شده است که به عنوان هدف عمل می کنند. برای هر مثال آموزشی، یک مجموعه از مقادیر ورودی (بردار ها) و یک یا چند مقدار خروجی تعیین شده مرتبط وجود خواهد داشت. هدف این شکل از آموزش داده ها این است که خطای طبقه بندی مجموع مدل ها را از طریق محاسبه صحیح مقدار خروجی مثال آموزشی از طریق آموزش، کاهش دهیم.

یادگیری بدون نظارت از این جهت متفاوت است که مجموعه آموزشی (training set) شامل هیچ برچسبی نمیباشد. موفقیت معمولاً این گونه تعیین میشود که آیا یک شبکه قادر به کاهش یا افزایش یک تابع هزینه (cost function) مرتبط خواهد بود یا نه. با این حال، مهم است که خاطر نشان کنیم اکثر کارهای تشخیص الگوی متمرکز بر تصاویر معمولاً به طبقه بندی با استفاده از یادگیری نظارت شده بستگی دارد.

شبکه های عصبی پیچشی (CNN) ها از این جهت که از نوروں هایی تشکیل شده اند که خودشان را از طریق یادگیری بهینه میکنند مشابه ANN های سنتی هستند. همچنان هر نوروں یک ورودی دریافت میکند و عملیاتی اجرا میکنند (مانند یک ضرب داخلی که بدنبال آن یک تابع غیرخطی می آید) که پایه و اساس ANN های بی شماری است. از بردار های تصویر خام ورودی تا خروجی نهایی که امتیاز طبقه است، کل شبکه همچنان یک تابع امتیاز ادراکی واحد (وزن) را بیان خواهد کرد. لایه آخر شامل توابع هزینه (loss function) مربوط به هر طبقه خواهد بود و تمام نکات و اصولی که برای ANN های سنتی توسعه یافته بودند همچنان اعمال می شوند.

تنها تفاوت قابل ذکر بین CNN ها و ANN های سنتی این است که CNN ها در درجه اول در حوزه تشخیص الگو درون تصاویر استفاده میشوند. این به ما اجازه می دهد که ویژگی های مختص به تصاویر را در معماری رمزگشایی کنیم که باعث میشود شبکه برای کارهای متمرکز بر تصاویر مناسب تر شود، ضمن اینکه پارامتر هایی که برای راه اندازی مدل نیاز است را کاهش میدهد.

یکی از بزرگترین محدودیت های شکل های سنتی ANN این است که آنها تمایل دارند با پیچیدگی های محاسباتی مورد نیاز برای داده های تصویر کامپیوتری تقلا کنند. دیتاست های رایج تمرینی یادگیری ماشین از قبیل دیتابیس ارقام دست نوشته MNIST برای بیشتر

اشکال ANN بدلیل ابعاد کوچک تصویر که فقط 28×28 است، مناسب هستند. با این دیتاست یک نورون در اولین لایه پنهان فقط شامل 784 وزن $28 \times 28 \times 1$ (که 1 به این دلیل است که MNIST فقط شامل مقادیر سیاه و سفید است) است که برای اکثر شکل های ANN قابل مدیریت است.

اگر یک تصویر رنگی 64×64 را در نظر بگیرید، تعداد وزن های یک نورون در اولین لایه به شکل قابل توجهی افزایش پیدا می کند و به 12,288 می رسد. همچنین توجه کنید که برای مدیریت این مقدار ورودی، شبکه هم باید خیلی بزرگ تر از شبکه ای که برای ارقام MNIST استفاده کردیم باشد، پس متوجه عیب های استفاده از چنین مدل هایی خواهید شد.

1.1 بیش برآزش (Overfitting)

ولی چرا برایمان مهم است؟ قطعاً ما میتوانیم فقط تعداد لایه های پنهان شبکه مان را افزایش دهیم، و احتمالاً تعداد نورون های درون آنها را افزایش دهیم. جواب ساده به این سوال نه است. به دو دلیل، اول از همه مشکل ساده نداشتن قدرت و زمان محاسبه نامحدود برای آموزش این ANN های عظیم.

دلیل دوم متوقف کردن یا کاهش دادن اثرات بیش برآزش است. **Overfitting** یا **بیش برآزش** وقتی رخ میدهد که بنا به چند دلیل، شبکه قادر به یادگیری موثر نباشد. این موضوع مهمی در تقریباً همه الگوریتم های یادگیری ماشین است و مهم است که تدابیر لازم برای کاهش تأثیرات آن را انجام دهیم. اگر مدل های ما نشانه هایی از بیش برآزش نشان دهند، ممکن است توانایی کاهش یافته ای در شناسایی ویژگی های عمومی نه تنها برای مجموعه داده های آموزشی، بلکه برای مجموعه های آزمایشی و پیش بینی نیز مشاهده کنیم.

این اصلی ترین دلیل کاهش پیچیدگی ANN هایمان است. هرچه پارامتر های کمتری برای آموزش نیاز باشد، احتمال این که شبکه ما بیش برآزش کند کمتر میشود و قطعاً منجر به بهبود عملکرد پیش بینی مدلمان میشود.

2 معماری CNN

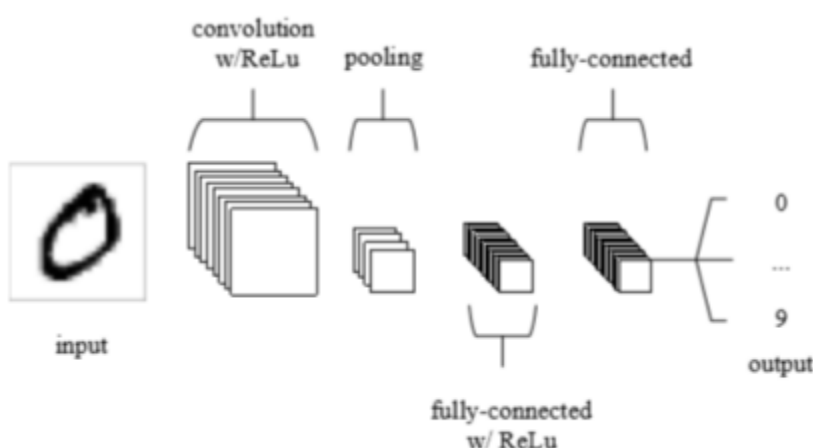
همانطور که قبلاً گفتیم CNN ها عمدتاً بر این تمرکز دارند که ورودی از تصاویر تشکیل شده باشد. این باعث میشود که معماری CNN ها بر این متمرکز شود که برای مدیریت نوع خاصی از داده ها مناسب باشد.

یکی از تفاوت های کلیدی آن این است که نورون های درون لایه های CNN از نورون هایی تشکیل شده اند که در سه بعد سازمان دهی شده اند: ابعاد فضایی ورودی (**طول** و **عرض**) و **عمق**. عمق به تعداد کل لایه های درون ANN اشاره ندارد، بلکه به بعد سوم یک حجم فعال سازی اشاره دارد. بر خلاف ANN های استاندارد، نورون های درون هر لایه داده شده، تنها به ناحیه کوچکی از لایه پیشین خود متصل میشوند.

در عمل این به این معنی است که در مثالی که پیش تر داده شد، 'حجم' ورودی ابعاد $64 \times 64 \times 3$ را خواهد داشت (طول، عرض و عمق)، که منجر به لایه خروجی نهایی ای میشود که از ابعاد $n \times 1 \times 1$ تشکیل شده باشد که n تعداد ممکن طبقه هاست، همانطور که ما تمام ابعاد ورودی را به حجم کوچکتری از امتیاز طبقه که در بعد عمق فشرده شده اند، ثبت کرده ایم.

2.1 معماری کلی

شبکه های عصبی پیچشی (CNN) ها از سه نوع لایه تشکیل شده اند: لایه های پیچشی، لایه های ادغام (pooling) و لایه های کاملاً همبند. وقتی این لایه ها بر روی هم انباشته میشوند، یک معماری CNN شکل میگیرد. یک معماری CNN ساده شده برای طبقه بندی MNIST در شکل 2 نشان داده شده است.



شکل 2: یک معماری CNN ساده شده، متشکل از فقط 5 لایه

عملکرد پایه CNN مثال بالا میتواند به چهار بخش کلیدی تقسیم شود:

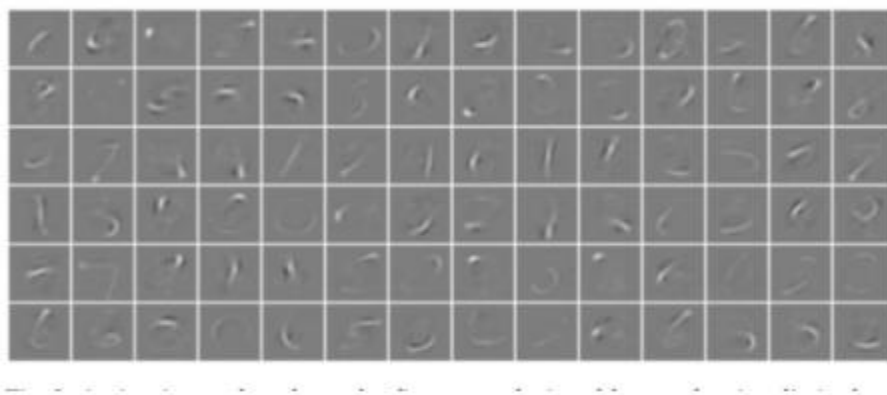
1. همانگونه که در شکل های دیگر ANN یافت میشود، **لایه ورودی** مقادیر پیکسل های تصویر را نگه میدارد.

2. **لایه کانولوشن** (پیچشی) خروجی نوروتهایی را تعیین میکند که به نواحی محلی ورودی متصل هستند، از طریق محاسبه ضرب اسکالر بین وزن هایشان و ناحیه متصل به حجم ورودی. **واحد یکسو شده خطی** (معمولاً به اختصار ReLU نامیده میشود) هدف دارد یک تابع فعالسازی عنصری مانند سیگموید را به خروجی فعالسازی تولید شده توسط لایه قبلی اعمال میکند.

3. **سیس لایه ادغام** (pooling) به سادگی نمونه برداری را در طول ابعاد فضایی ورودی داده شده انجام میدهد و تعداد پارامتر ها را در آن فعالسازی کاهش میدهد.

4. سپس **لایه های کاملاً همبند** همان وظایفی که در ANN های استاندارد یافت میشود انجام میدهند و تلاش میکنند امتیاز های طبقه را از فعالسازی ها تولید کنند تا برای طبقه بندی استفاده شوند. همچنین توصیه شده است که برای بهبود عملکرد، ReLU میتواند بین این لایه ها استفاده شود.

با این روش تبدیل ساده، CNN ها قادرند ورودی اصلی را به صورت لایه به لایه با استفاده از تکنیک های پیچشی و نمونه برداری تبدیل کنند تا برای اهداف طبقه بندی و رگرسیون، امتیاز طبقه ها را تولید کنند.



شکل 3: فعالسازی های گرفته شده از اولین لایه پیچشی یک CNN عمیق ساده، پس از آموزش دیتابیس ارقام دست نویس MNIST. اگر با دقت نگاه کنید میتوانید ببینید که شبکه با موفقیت ویژگی های منحصر بفرد رقم به خصوص را تشخیص داده است.

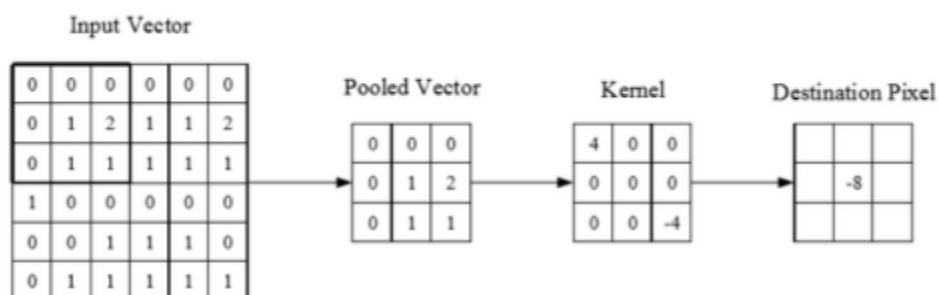
با این حال، مهم است به یاد داشته باشیم که درک معماری کلی یک CNN کافی نیست. ساخت و بهینه سازی این مدل ها میتواند زمانگیر و گیج کننده باشد. حال میخواهیم هر لایه، هاپیرپارامتر ها و اتصالاتشان را با جزئیات بررسی کنیم.

2.2 لایه کانولوشن (پیچشی)

همانگونه که از اسم آن پیداست، لایه کانولوشن نقشی حیاتی در این که چگونه شبکه های عصبی پیچشی (CNN ها) عمل میکنند، دارد. پارامترهای این لایه بر روی استفاده از **کرنل (هسته)** های قابل یادگیری تمرکز دارند.

این کرنل ها معمولاً ابعاد فضایی کوچکی دارند، ولی در طول تمام عمق ورودی پخش میشوند. وقتی داده به لایه کانولوشن میرسد، این لایه هر فیلتر را در طول ابعاد فضایی ورودی می پیچد تا یک نقشه (map) فعالسازی دوبعدی تولید کند. همانگونه که در شکل 3 میبینید، این نقشه ها می توانند به تصویر در آیند.

به طور کلی، در حینی که ما از ورودی عبور می‌کنیم، حاصل ضرب اسکالر برای هر مقدار در آن هسته محاسبه می‌شود (شکل 4). از این طریق، شبکه یاد می‌گیرد که هسته‌هایی را ایجاد کند که هنگام مشاهده یک ویژگی خاص در موقعیت فضایی معین ورودی، فعال شوند. این‌ها به طور معمول با نام **فعال‌سازی‌ها** شناخته می‌شوند.



شکل 4: یک نمایش تصویری از لایه کانولوشن. عنصر مرکزی کرنل بر روی بردار ورودی قرار می‌گیرد، که بعداً محاسبه شده و با مجموع وزن دار خودش و پیکسل‌های اطرافش جایگزین شده.

هر کرنل یک نقشه فعال‌سازی متناظر خواهد داشت که در طول بعد عمق بر روی هم قرار گرفته تا حجم خروجی کامل از لایه کانولوشن را تشکیل دهند.

همانگونه که قبلاً اشاره کردیم، آموزش ANN‌ها با ورودی‌هایی از قبیل تصاویر منجر به مدل‌هایی می‌شود که برای آموزش موثر بیش از حد بزرگ هستند. این به رفتار کاملاً همبند نورون‌های ANN‌های استاندارد برمیگردد. برای بهتر کردن این وضع، همه نورون‌های لایه کانولوشن تنها به ناحیه کوچکی از حجم ورودی متصل است که معمولاً به آن **اندازه میدان پذیرنده** نورون گفته می‌شود. بزرگی اتصال در طول عمق تقریباً همیشه برابر عمق ورودی است.

برای مثال، اگر ورودی شبکه یک تصویر $3 \times 64 \times 64$ باشد (یک تصویر رنگ شده RGB به ابعاد 64×64) و اندازه زمینه پذیرنده 6×6 باشد، ما در مجموع 108 وزن بر هر نورون در لایه کانولوشن داریم. ($3 \times 6 \times 6$ که 3 بزرگی اتصال در عمق حجم است)، برای مقایسه یک نورون استاندارد در سایر شکل‌های ANN شامل 12,288 وزن خواهد بود.

لایه‌های کانولوشن همچنین قادر هستند پیچیدگی مدل را از طریق بهینه‌سازی خروجی به شکل قابل توجهی کاهش دهند. آنها از طریق سه هاپیر پارامتر بهینه می‌شوند: **عمق**، **گام** و **تنظیم حاشیه صفر**.

عمق حجم خروجی تولید شده توسط لایه‌های کانولوشن را می‌توان به صورت دستی از طریق تعداد نورون‌های موجود در این لایه به همان ناحیه ورودی تنظیم کرد. این می‌تواند در سایر شکل‌های ANN‌ها هم دیده شود، جایی که همه نورون‌های لایه پنهان بطور

مستقیم به تک تک نورون های قبلی متصل است. کاهش این هاپیر پارامتر میتواند مقدار قابل توجهی مجموع تعداد نورون های یک شبکه را کم کند. اما همچنین میتواند قابلیت های تشخیص الگوی مدل را هم به طرز قابل توجهی کم کند .

همچنین ما قادر هستیم گامی را تعریف کنیم که در آن عمق را در اطراف ابعاد فضایی ورودی تنظیم کنیم تا میدان پذیرنده را قرار دهیم. مثلاً اگر گام را برابر 1 قرار دهیم، یک میدان پذیرنده به شدت روی هم افتاده و دارای اشتراک خواهیم داشت که فعالسازی های بسیار بزرگی تولید میکند. اگر در عوض گام را عددی بزرگتر از 1 قرار دهیم اشتراکات را کاهش داده و خروجی ای با ابعاد فضایی کمتر تولید میکند.

حاشیه گذاری صفر (zero-padding) فرآیند ساده حاشیه گذاری مرز ورودی است و روش موثری برای افزایش کنترل روی ابعاد حجم های خروجی است .

مهم است که درک کنیم از طریق استفاده از این تکنیک ها ما ابعاد فضایی خروجی لایه های کانولوشن را تغییر میدهیم. برای محاسبه آن میتوانید از فرمول زیر استفاده کنید:

$$\frac{(V - R) + 2Z}{S + 1}$$

که V اندازه حجم ورودی (طول \times عرض \times عمق)، R اندازه میدان پذیرنده، Z مقدار حاشیه صفر و S به گام اشاره دارد. اگر نتیجه محاسبه شده یک عدد صحیح نباشد، گام اشتباه تنظیم شده است، زیرا نورون ها قادر نخواهند بود به تمیزی در ورودی داده شده هم خوانی داشته باشد .

با وجود تلاش های زیادمان، اگر از تصویری با ابعاد واقعی استفاده کنیم، مدل هایمان همچنان بسیار بزرگ خواهند بود. با این حال روش هایی ساخته شده اند که تعداد کل پارامتر های درون لایه های کانولوشن را بسیار کاهش دهند.

اشتراک گذاری پارامتر ها با این فرض عمل میکند که اگر یک ویژگی ناحیه در یک مجموعه ناحیه فضایی در محاسبه مفید است، احتمالاً در یک ناحیه دیگر هم مفید خواهد بود. اگر هر نگاشت فعالسازی در حجم خروجی را به همان وزن و بایاس محدود کنیم، کاهش شدیدی را در تعداد پارامتر های تولید شده توسط لایه کانولوشن خواهیم دید.

در نتیجه ، همزمان با رخ دادن مرحله انتشار معکوس (backpropagation) هر نورون در خروجی نمایانگر گرادینان کلی خواهد بود که می‌تواند در عمق جمع شود، در نتیجه بجای تک تک وزن ها فقط یک مجموعه وزن آپدیت میشوند.

2.3 لایه ادغام (پولینگ)

هدف لایه های ادغام این است که ابعاد نمونه را به تدریج کاهش دهد و در نتیجه منجر به کاهش تعداد پارامتر ها و پیچیدگی محاسباتی مدل شود.

لایه ادغام روی هر نقشه فعالسازی در ورودی عمل میکند و ابعاد آن را با استفاده از تابع MAX توزین میکند. در اکثر CNN ها، این لایه ها به شکل max-pooling هستند با کرنل (هسته) هایی با ابعاد 2×2 با گام 2 که روی ابعاد فضایی ورودی اعمال میشود. این کار نقشه فعالسازی را به 25% اندازه اصلی اش میرساند، ضمن اینکه عمق حجم را به اندازه استانداردش نگه میدارد.

بدلیل ذات تخریبگر لایه ادغام، فقط دو روش max-pooling معمولاً دیده میشود. معمولاً گام ها و فیلترهای لایه های پولینگ به 2×2 تنظیم میشود، که به لایه اجازه می‌دهد در تمام ابعاد فضایی ورودی گسترش یابد. به علاوه پولینگ دارای اشتراک ممکن است استفاده شود، که گام به 2 و سائز کرنل به 3 تنظیم میشود. بدلیل ذات تخریبگر پولینگ، داشتن سائز کرنل بالای 3 معمولاً عملکرد مدل را به شدت کاهش میدهد.

همچنین مهم است درک کنیم که فرای max-pooling ، معماری های CNN ممکن است شامل پولینگ کلی باشند. لایه های پولینگ کلی از نورون های پولینگ تشکیل شده اند که قادرند چند عمل رایج از جمله نرمال سازی L1 و L2 و پولینگ میانگین اجرا کنند. با این حال این آموزش، بیشتر بر استفاده از max-pooling تمرکز خواهد کرد.

2.4 لایه کاملاً همبند

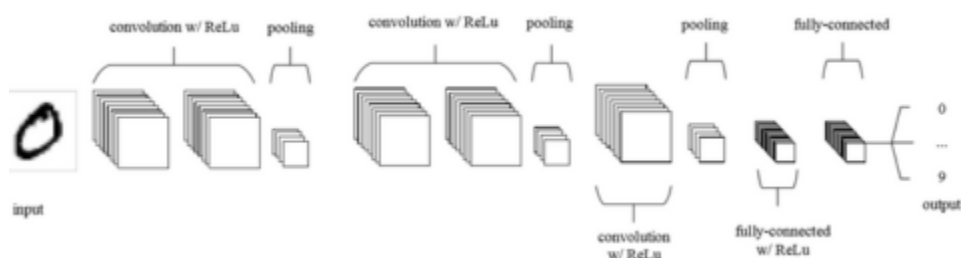
لایه کاملاً همبند شامل نورون هایی است که مستقیماً به دو لایه همسایه اش متصل اند، بدون این که به هیچ لایه ای درون آنها متصل باشند. این مشابه طرز قرار گرفتن نورون ها در شکل های سنتی ANN ها هستند. (شکل 1)

3 دستور تهیه

با وجود لایه های نسبتاً کمی که برای شکل گیری CNN نیاز است، هیچ راهی برای فرموله کردن یک معماری CNN وجود ندارد. بنابراین احتمالاً خواهد بود اگر به سادگی چند لایه را باهم بیندازیم و انتظار داشته باشیم کار کند. از طریق خواندن ادبیات مرتبط، واضح است که مثل سایر اشکال ANN ها، CNN ها تمایل دارند یک معماری مشترک را دنبال کنند. این معماری مشترک که در

شکل 2 به تصویر کشیده شده است، لایه های کانولوشن روی هم قرار میگیرند و به دنبال آن لایه های ادغام در حالتی تکرار شونده به خورد لایه های کاملاً همبند داده میشوند .

یک معماری CNN رایج دیگر این است که دو لایه کانولوشن قبل از یک لایه پولینگ (ادغام) قرار گیرد که در شکل 5 میبینیم. این روش به شدت توصیه میشود زیرا روی هم جمع کردن چندین لایه کانولوشن اجازه انتخاب شدن ویژگی های پیشرفته بیشتری از ورودی را میدهد .



شکل 5: یک شکل رایج از معماری CNN که در آن لایه های کانولوشن بین ReLU ها پشت سر هم قرار گرفته اند، قبل از ورود به لایه پولینگ و قبل از رفتن بین یک یا چند ReLU کاملاً همبند.

همچنین توصیه میشود که لایه های کانولوشن بزرگ را به چندین لایه کانولوشن کوچکتر تقسیم کنیم. این کار برای کاهش پیچیدگی محاسباتی درون هر لایه کانولوشن داده شده است. مثلاً اگر میخواهید سه لایه کانولوشن با میدان پذیرنده 3×3 روی هم قرار دهید. هر نورون لایه کانولوشن اول، دید 3×3 از بردار ورودی خواهد داشت. پس یک نورون در لایه کانولوشن دوم دید 5×5 از بردار ورودی خواهد داشت و یک نورون در لایه کانولوشن سوم دید 7×7 از بردار ورودی خواهد داشت. چون این پشته ها غیرخطی هستند به ما اجازه میدهد که ویژگی های قوی تر ورودی را با پارامتر های کمتری بیان کنیم. با این حال مهم است درک کنیم که این کار نیاز به اختصاص حافظه مجزا دارد، مخصوصاً هنگام استفاده از الگوریتم انتشار معکوس (backpropagation)

لایه ورودی باید بطور بازگشتی بر 2 بخش پذیر باشد. اعداد رایج شامل 32×32 ، 64×64 ، 96×96 ، 128×128 و 224×224 هستند .

ضمن استفاده از فیلتر های کوچک، گام را به یک تنظیم کنید و از حاشیه صفر استفاده کنید تا مطمئن شوید که لایه های کانولوشن هیچ یک از ابعاد ورودی را دوباره پیکربندی نمیکنند. مقدار حاشیه صفر استفاده شده باید با کم کردن یک از اندازه میدان پذیرنده و تقسیم آن بر دو حساب شود .

شبکه های عصبی کانولوشن (CNN ها) الگوریتم یادگیری ماشین بسیار قدرتمندی است، با این حال آنها می توانند به طرز وحشتناکی سنگین-منبع باشند. یک مثال از این مشکل می تواند در فیلتر کردن یک تصویر بزرگ باشد (هر تصویری بیشتر از 128×128 بزرگ محسوب می شود). پس اگر تصویر 227×227 باشد (همانگونه که در دیتاست ImageNet دیده می شود) و ما با 64 کرنل (هسته) ورودی که هر کدام دارای یک حاشیه (پدینگ) صفر هستند، نتیجه سه بردار فعالسازی به اندازه $64 \times 227 \times 227$ خواهد بود که حدودا 10 میلیون فعالسازی یا حافظه عظیم 70 مگابایتی برای هر عکس می شود. در این صورت شما دو انتخاب دارید. اول، می توانید ابعاد فضایی تصاویر ورودی را با تغییر سایز تصاویر به چیزی سبک تر، کوچک کرده. بجای این کار، شما می توانید خلاف هر آنچه ما در این مقاله بیان کردیم بروید و اندازه فیلترهای بزرگتر با یک گام بزرگتر (2 بجای 1) انتخاب کنید.

علاوه بر چند قانون کلی که در بالا ذکر شده است، مهم است که چند ترفند را هم درباره تکنیک های آموزش کلی ANN ها یاد بگیرید. نویسندگان توصیه می کنند مقاله فوق العاده جفری هینتون با عنوان A Practical Guide to Training Restricted Boltzmann Machines را بخوانید.

4 نتیجه گیری

شبکه های عصبی پیچشی از این جهت با سایر شکل های شبکه های عصبی مصنوعی متفاوت هستند که بجای تمرکز بر روی تمام دامنه مسئله، از دانش بر روی نوع خاصی از ورودی به کار گرفته می شود. که منجر می شود معماری شبکه ساده تری راه اندازی شود.

این مقاله مفاهیم پایه شبکه های عصبی پیچشی را توضیح داده است، لایه های مورد نیاز برای ساخت یک شبکه را شرح داده و نحوه بهترین ساختاردهی شبکه را در اکثر کارهای تحلیل تصویر بیان کرده است.

در سال های اخیر، تحقیقات در حوزه تحلیل تصویر با استفاده از شبکه های عصبی تقریبا آرام بوده است که بیشتر بدلیل باور های نادرست پیرامون سطح پیچیدگی و دانش مورد نیاز برای شروع مدلسازی این الگوریتم های یادگیری ماشین بسیار قدرتمند است. نویسندگان امیدوارند که این مقاله به نحوی این سردرگمی ها را کاهش داده و این حوزه را برای مبتدیان در دسترس تر کرده باشد.