تشخیص اعداد و حروف الفبا انگلیسی به وسیله شبکه های عصبی کانولوشن و الگوریتم ماشین لرنینگی نزدیک ترین همسایگی

معماری شبکه عصبی کانولوشن CNN

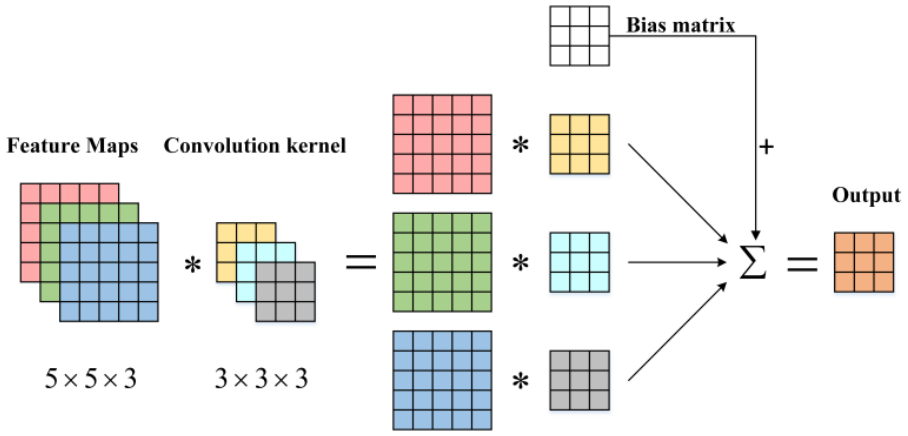
طبقه‌بندی تصویر به عنوان یک موضوع تحقیقاتی کلاسیک در سال‌های اخیر، یکی از موضوعات اصلی بینایی رایانه و اساس حوزه‌های مختلف تشخیص بصری است. بهبود عملکرد شبکه طبقه بندی به طور قابل توجهی سطح کاربرد آن را بهبود می بخشد. بهبود فناوری طبقه‌بندی تصویر بخش مهمی از ارتقاء توسعه بینایی رایانه است. فرآیند اصلی آن شامل پیش پردازش داده های تصویری ، استخراج و نمایش ویژگی ها [1] و طراحی طبقه بندی کننده است. تمرکز تحقیقات طبقه بندی تصاویر همیشه استخراج ویژگی های تصویر بوده است که اساس طبقه بندی تصاویر است. الگوریتم‌های سنتی استخراج ویژگی تصویر بیشتر بر روی تنظیم دستی ویژگی‌های خاص تصویر تمرکز می‌کنند. ANN یک شبکه عصبی بیولوژیکی انتزاعی است که یک مدل عملیات ریاضی میباشد که از تعداد زیادی نورون به هم پیوسته تشکیل شده است. تقریباً پردازش شبکه عصبی سیگنال های عصبی را شبیه سازی می کند. در ابتدا مک کالوچ و پیتس شبکه های عصبی بیولوژیکی را تجزیه و تحلیل کردند و یک مدل ریاضی عملیات منطقی داخلی از فعالیت نورون – مدل نورون MP را پیشنهاد کردند.

روزنبلات توابع یادگیری را به مدل MP اضافه کرد و یک مدل پرسپترون تک لایه را پیشنهاد کرد و برای اولین بار تحقیق در مورد شبکه های عصبی را عملی کرد. پس از آن، هوبر و ویز و همکاران قشر بینایی مغز گربه را مطالعه کردند و دریافتند که نورون های بینایی بیولوژیکی اطلاعات را بر اساس تحریک منطقه ای محلی درک می کنند، آنها به این نتیجه رسیدند که ادراک بصری لایه به لایه از طریق میدان های گیرنده چند سطحی تحریک می شود. بعدها، محققان سعی کردند از پرسپترون چندلایه برای یادگیری ویژگی‌ها استفاده کنند و مدل را با الگوریتم پس انتشار (BP) آموزش دادند .این کشف الهام بخش محققان برای ساخت یک شبکه عصبی کامپیوتری شبیه به یک سیستم بینایی بیولوژیکی شد و CNN متولد شد. لکون و همکاران اولین دسته از مدل CNN-LeNet-5 را ارائه کرد .با این حال، به دلیل فقدان داده‌های آموزشی در مقیاس بزرگ، آن نیز توسط مبانی نظری و قدرت محاسباتی رایانه محدود شد . از سال 2017 تا به امروز، مدل های بیشتری با عملکرد برتر یکی پس از دیگری ظاهر شده اند. CNN ها به طور فزاینده ای برتری غیرقابل جایگزینی را در طبقه بندی تصاویر نشان داده اند . به طور کلی، روش‌های طبقه‌بندی صحنه تصویر سنجش از دور مبتنی بر CNN را می‌توان به سه نوع تقسیم کرد: 1- مدل های از پیش آموزش‌دیده‌شده به عنوان استخراج‌کننده ویژگی استفاده می‌شوند [3-4]. 2- مدل های از پیش آموزش دیده را روی مجموعه داده تنظیم دقیق کنید [4-5-6]. 3- وزن CNN ها را برای آموزش در سطح جهانی مقداردهی کنید [7-8-9].

ConvNet قادر است به طور موفقی وابستگی‌های زمانی و فضایی را در یک تصویر با استفاده از فیلترهای مرتبط ثبت کند و همچنین، معماری فیلترگذاری بهتری را روی مجموعه داده تصویر به دلیل کاهش تعداد پارامترهای درگیر و استفاده مجدد از وزن‌ها انجام می‌دهد. به بیان دیگر، شبکه می‌تواند برای درک تصاویر پیچیده به طور بهتری آموزش ببیند. برای CNN هایی با عمق معین، عملیات کانولوشن چندین لایه پیچیدگی می تواند ویژگی های مختلف ورودی را استخراج کند. کانولوشن لایه پایین به طور کلی ویژگی های مشترک مانند بافت،

خطوط و لبه ها را استخراج می کند، در حالی که لایه بالاتر ویژگی های انتزاعی بیشتری را استخراج می کند. لایه کانولوشن دارای چندین هسته کانولوشن است.

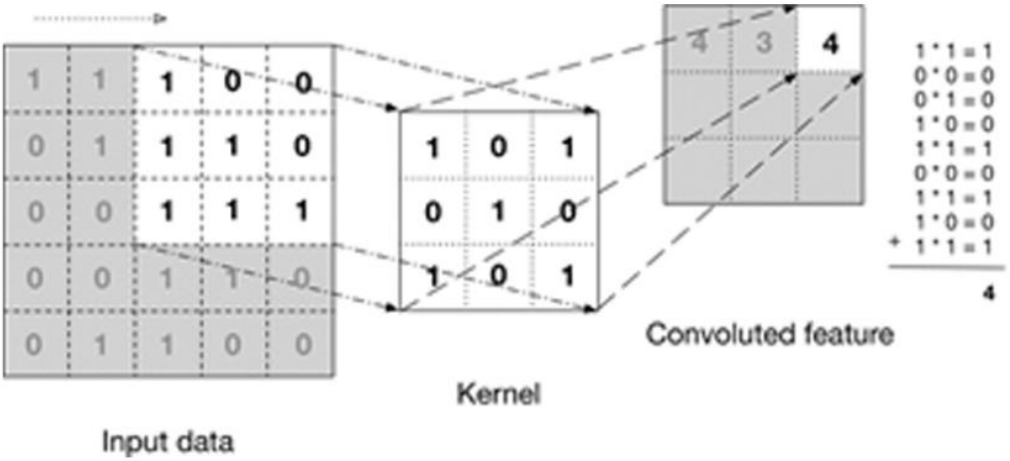
در تصاویر سیاه و سفید با توجه به میزان رنگ هر پیکسل، هر عنصر از ماتریس مقدار دهی (0 تا 255) می‌شود. یک تصویر RGB چیزی نیست جز ماتریسی از مقادیر پیکسلی که دارای سه صفحه باشد. در حالی که یک تصویر در مقیاس خاکستری دارای یک صفحه است. برای درک بیشتر به این تصویر نگاهی بیندازید. تصاویر رنگی ساختار به شکل زیر دارند و تصویر رنگی از سه صفحه تشکیل شده است که 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 0 تا 255 است. از ترکیب سه عدد، یک رنگ نهایی حاصل می‌شود. بنابراین هر تصویر رنگی معادل 3 ماتریس است که هر کدام برای یکی از رنگ‌ها در نظر گرفته می‌شود .



سازو کار هر شبکه ی عصبی به این صورت است که یک سری بردار ورودی دریافت می‌کند و سپس اطلاعات از چندین لایه ی مخفی عبور داده می‌شود. هر لایه مخفی شامل چندین نرون است و نرون‌های هر لایه به صورت مستقل عملیات وزن دهی را انجام می‌دهند و به نرون‌های لایه ی قبل از خود متصل هستند و در نهایت نتیجه در لایه خروجی که یک لایه ی کاملا متصل است مشاهده می‌شود. هر لایه در ساختار شبکه ی کانولوشن یک جز اساسی است و کاری مجزا از دیگر لایه ها را انجام می‌دهد که این عملیات ترکیبی از عملیات خطی و غیر خطی می‌باشد که این عملیات ها شامل کانولوشن و تابع فعالسازی است که منجر به استخراج ویژگی می‌شود.

ایده ی اصلی شبکه عصبی کانولوشن عملیات کانولوشن یا پیچش است. انجام پروسه کانولوشن به طور کلی به این صورت است که در این شبکه یک عملگر کانولوشن و یک کرنل یا فیلتر کانولوشن وجود دارد. عملگر کانولوشن، فیلتر کانولوشن را برمی‌دارد و بر روی تصویر حرکت می‌دهد به نحوی که در ماتریس 6*6 ابتدا مشاهده می‌کنید؛ فیلتر ابتدا هرسطر 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 و تصویر وردی اسکن می‌شود، و توسط عملگر کانولوشن عملیات فیلترینگ انجام می‌شود و نتیجه در ماتریس خروجی نشان داده می‌شود.

برای درک بهتر نحوه کار شبکه عصبی کانولوشن روال کار با تصاویر در مقیاس خاکستری را بررسی کنیم. تصویر زیر نشان می‌دهد که شبکه عصبی کانولوشن CNN چیست؟ ما یک فیلتر / هسته (ماتریس 3 × 3) در نظر می‌گیریم و آن را روی تصویر ورودی اعمال می-کنیم تا ویژگی را بدست آوریم. این ماتریس ویژگی به لایه بعدی منتقل می‌شود.



لایه های شبکه عصبی کانولوشن

شبکه های عصبی کانولوشن از چندین لایه نورون مصنوعی تشکیل شده اند. نورون های مصنوعی، تقلیدی تقریبی از نمونه های بیولوژیکی آنها و بر اساس توابع ریاضیاتی هستند که مجموع وزنی چندین ورودی و خروجی را با تابع فعال ساز محاسبه می‌کنند. هنگام وارد کردن تصویر در شبکه ی عصبی کانولوشن، هر لایه چندین عملکرد فعال‌سازی ایجاد می‌کند که به لایه بعدی منتقل می‌شوند. لایه اول معمولاً ویژگی‌های اساسی مانند لبه های افقی یا مورب را استخراج می‌کند. این خروجی به لایه بعدی منتقل می‌شود که ویژگی‌های پیچیده تری مانند گوشه ها یا لبه های ترکیبی را تشخیص می‌دهد.

بر اساس نقشه فعال‌سازی لایه کانولوشن نهایی، لایه طبقه بندی مجموعه ای از مقادیر (مقادیر بین 0 و 1) را تولید می‌کند که مشخص می-کند احتمال تعلق تصویر به یک "کلاس" چیست. به عنوان مثال ، اگر یک کانولوشن دارید که گربه ها، سگ ها و اسب ها را تشخیص می-دهد، با دادن تصویر هریک از این حیوانات انتظار می‌رود شبکه به درستی تشخیص دهد که تصویر متعلق به چه گروهی است.

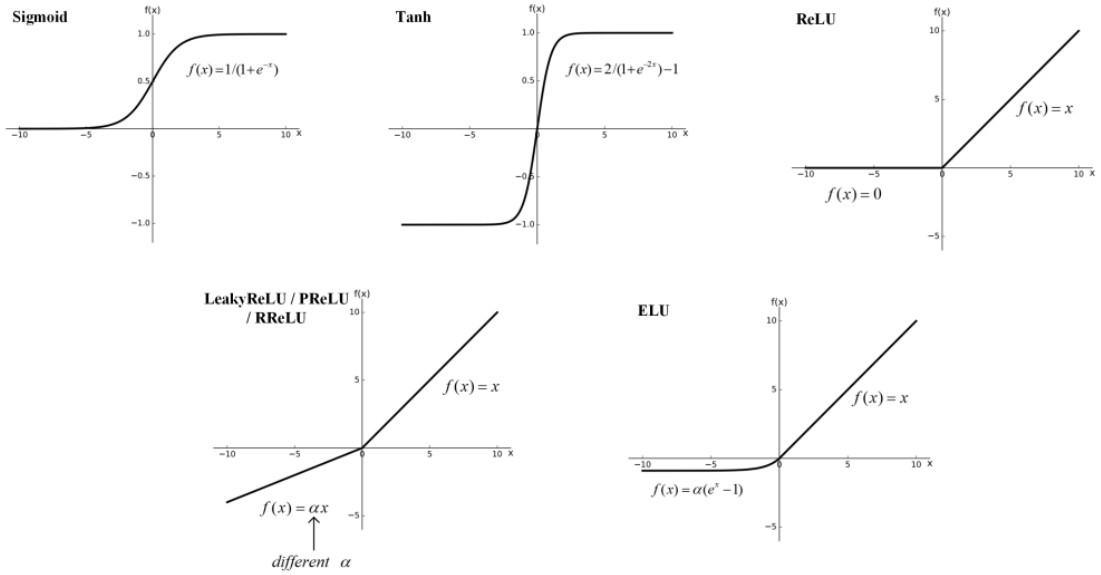
لایه تجمعی (Pooling)

همانند لایهConvolutional ، لایه Pooling وظیفه کاهش اندازه فضایی ویژگی Convolved را دارد. این کار برای کاهش توان محاسباتی مورد نیاز برای پردازش داده ها با کاهش ابعاد است. دو نوع pooling متوسط و حداکثر وجود دارد. بنابراین آنچه که در Poolingانجام می‌شود این است که حداکثر مقدار پیکسل را از بخشی از تصویر که پوشانده شده توسط هسته پیدا می‌کنیم Pooling .

همچنین به عنوان یک تعدیل کننده نویز عمل می‌کند و کاهش نویز را به همراه کاهش ابعاد انجام می‌دهد. در انتهای هر مرحله ی بعد از استخراج ویژگی برای کاهش ابعاد pooling انجام می‌شود. هر لایه ی شبکه ی عصبی شامل یک لایه ی Convolutional است ویک لایه Pooling است. با بالا رفت این لایه ها قادر به ثبت جزئیات بیشتری خواهیم بود ولی باید در نظر داشته باشیم که افزایش لایه ها منجر به افزایش هزینه ی محاسبات خواهد شد. تعادل بین عمق، دقت و سرعت شبکه بسیار ساده است. هر چه لایه‌های بیشتری استفاده شوند، دقت کلی بالاتر می‌رود؛ اما باعث می‌شود که شبکه کندتر اجرا شود، زیرا تعداد محاسبات افزایش می‌یابد. به طور کلی، می‌توان عمق‌های مختلف را بررسی کرد و بهترین نقطه که بالاترین نتیجه را به دست می‌دهد را یافت. در برخی موارد باید آگاه بود که تعادل بین دقت و سرعت تحت قاعده «بازده نزولی» است یعنی هر چه لایه‌های بیشتری اضافه بکنیم، دقت کمتری در هر لایه منفرد اضافه می‌شود.

توابع فعالسازی غیر خطی

تابع فعال سازی این است که ورودی و خروجی را دارای یک رابطه عملکردی می کند که سیستم غیرخطی را وارد شبکه عصبی می کند و داشتن یک تابع فعال سازی غیرخطی مناسب می تواند به طور قابل توجهی عملکرد شبکه را بهبود بخشد [10]. شکل زیر چندین عملکرد فعال سازی رایج را نشان می دهد. از این میان، سیگموئید و تن را غیرخطی های اشباع کننده می نامند. از شکل 6 و تعریف فرمول می توان دریافت که وقتی ورودی بسیار بزرگ یا بسیار کوچک است، تابع Sigmoid در خروجی 0 یا 1 اشباع می شود و تابع Tanh در خروجی 1- یا 1 اشباع می شود. مشکلات ناشی از غیرخطی های اشباع، غیرخطی های غیراشباع مانند [11] ReLU، Leaky ReLU [12]، [13] PReLU، RReLU [14] و ELU [15] پیشنهاد شده است. از نظر زمان مورد نیاز برای آموزش نزول گرادیان، توابع اول بسیار کندتر از دومی هستند، نورون هایی با غیرخطی های غیر اشباع، واحدهای خطی اصلاح شده (ReLUs) نامیده می شوند. این نوع شبکه عصبی کانولوشن عمیق با ReLUs چندین برابر سریعتر از شبکه های مشابه با tanh به عنوان توابع فعال سازی است.



لایه کاملا متصل (FC).

لایه FC عموماً در پشت لایه پیوسته کانولوشن و لایه ادغام قرار دارد و نورون‌های بین لایه‌های مختلف آن کاملاً به هم متصل هستند. این اطلاعات محلی را با تبعیض طبقه بندی استخراج شده پس از کانولوشن و ادغام [16] ادغام و طبقه بندی می کند و در نهایت اطلاعات دسته بندی تصویر را خروجی می کند. آن شامل چندین لایه پنهان که ویژگی های سطح بالا را از شبکه قبلی به شکل پیچیده تری استخراج می کنند [15]. تعداد نورون ها در انتهای خروجی تعداد دسته ها است و سپس بردار خروجی برای تعیین اینکه تصویر متعلق به کدام دسته است استفاده می شود. به زبان ساده، FC Layer به عنوان یک طبقه بندی کننده در CNN عمل می کند. تحت آموزش شبکه، خروجی شبکه به طور کلی در معرض رگرسیون softmax [10] برای عادی سازی احتمال قبل از عملکرد از دست دادن لایه FC قرار می گیرد. البته Cu و همکاران. [10] همچنین تأثیر توابع از دست دادن چندگانه بر عملکرد شبکه را تحلیل کرد. پارامترهای لایه FC با استفاده از پس انتشار گرادیان به روز می شوند. هنگامی که یک مدل بزرگ با پارامترهای بیشتر بر روی یک مجموعه داده کوچکتر آموزش داده می شود، لایه FC به طور کلی از تنظیم و حذف L2 استفاده می کند. هدف اساسی از استفاده از آنها جلوگیری از برازش بیش از حد مدل است. مدل کلاسیک CNN اساساً از روش ترک تحصیل ReLU plus استفاده می کند و به عملکرد طبقه بندی خوبی دست یافته است.

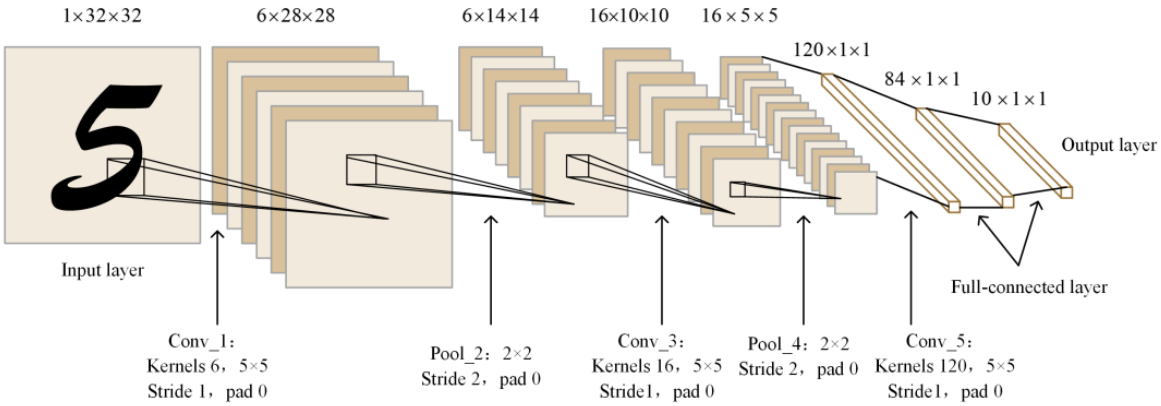
Loss Function

علاوه بر انواع لایه‌های مختلف معماری CNN که در بخش قبل معرفی شد، طبقه‌بندی نهایی از لایه خروجی معمولاً آخرین لایه لایه FC به دست می‌آید، همانطور که در شکل زیرنشان داده شده است. توابع تلفات مختلف نیز بر عملکرد CNN تأثیر می‌گذارند. معماری و برای کارهای بصری مختلف (به عنوان مثال، طبقه بندی تصویر، تشخیص چهره، و تشخیص اشیا) اعمال می شود .

Loss Function	Equation	Characteristic
L1 (MAE)	$Loss(y, y^*) = \frac{1}{m} \times \sum_{i=1}^{m} \left\| y_i^* - y_i \right\|$	This function is widely used in regression problems. L1 Loss is called mean absolute error (MAE)
L2 (MSE)	$Loss(y, y^*) = \frac{1}{m} \times \sum_{i=1}^{m} (y_i^* - y_i)^2$	This function is widely used in regression problems. L2 Loss is called mean square error (MSE)
Softmax + Cross-Entropy	$Loss(y, y^*) = -\sum_i \frac{y_i}{\sum_{i=1}^{m} y_i} \log(y_i^*), i \in [1, m]$	This function usually employed as a substitution of the MSE in multi-class classification problems. It is also commonly used in CNN models

Optimizer

جریان داده در معماری CNN اساساً در بخش فوق معرفی شده است. ما به وضوح درک می کنیم که آموزش شبکه بر مرحله اصلی به روز رسانی گرادیان متکی است، یعنی باید گرادیان تابع هدف (تابع ضرر) را با اعمال یک مشتق مرتبه اول با توجه به پارامترهای شبکه و سپس محاسبه کند. اطلاعات گرادیان در قالب محاسبه دیفرانسیل جزئی به لایه شبکه قبلی منتقل می شود تا به روز رسانی پارامترهای یادگیری هر لایه شبکه حاصل شود .



استفاده از مدل شبکه عصبی کانولوشنی برای تشخیص تصاویر

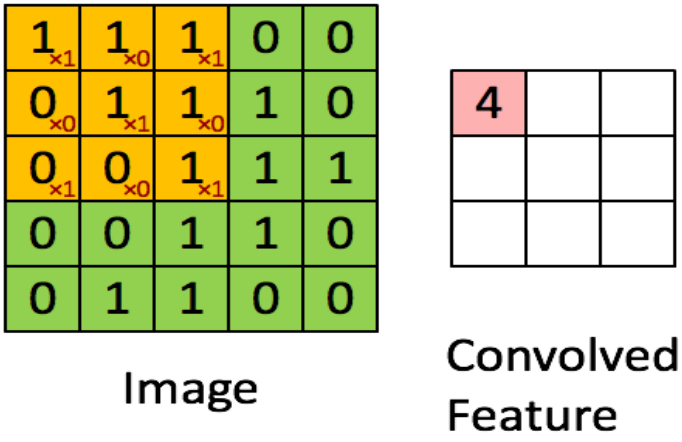
حال ما به بررسی دیتا ست EMNIST که شامل اعداد 0 تا 9 و حروف کوچک و بزرگ الفبای انگلیسی است میپردازیم .

پیش پردازش داد ها

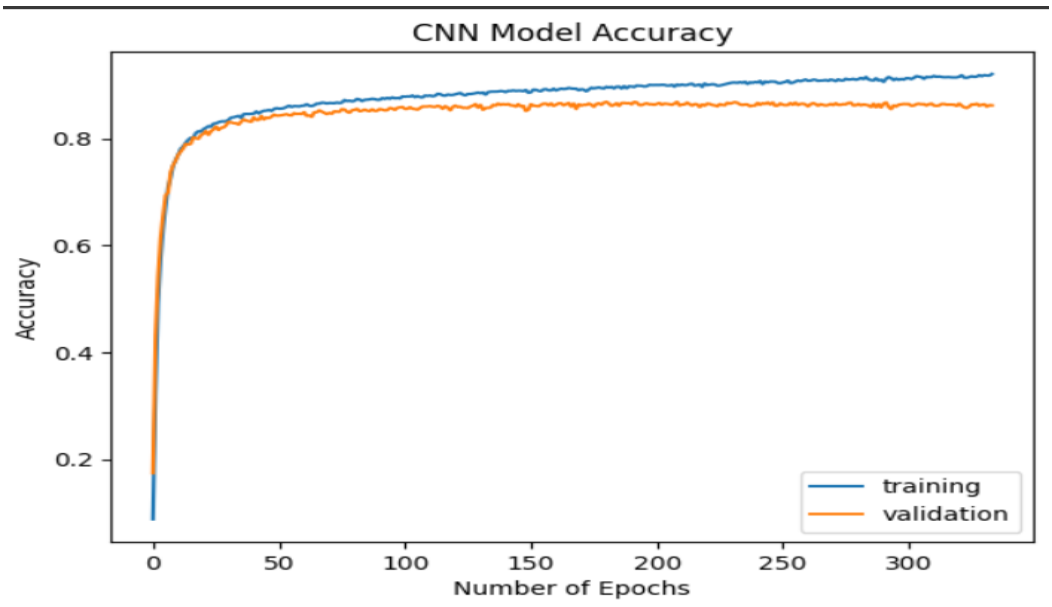
در این مرحله ابتدا لیبل ها را از عدد به حروف تبدیل میکنیم و تمام مقدار تصاویرو به 255 تقسیم کرده تا نرمال سازی صورت بگیرد و پس از تقسیم دیتاست به سه قسمت ترین و تست و اعتبارسنجی آنگاه آنگاه شروع به ساختن مدل میکنیم .

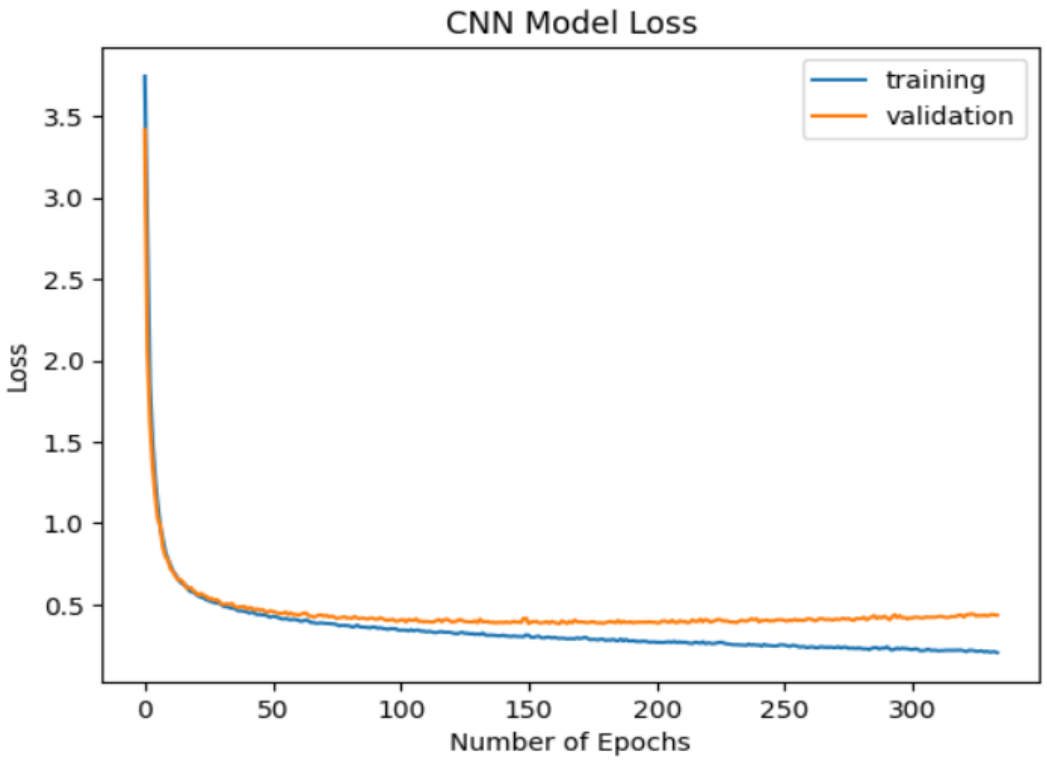
پیاده سازی معماری CNN

با استفاده از کتابخانه تنسورفلو عکس ها را به سایز 28*28 تغییر اندازه میدهیم ابتدا یه لایه کانولوشنی با 8 فیلتر 3*3 روی عکسامون اعمال میکنیم با تابع فعالسازی relu که سایز عکس به 26*26 کاهش میابد به دلیل اعمال فیلتر و بعد از لایه MaxPooling2D استفاده کرده که نیوزها رو کاهش داده و باعث کاهش سایز عکس میشود ولی طول لایه های آن افزایش یافته است .



پس از آن به تعدادی مناسب لایه های ConvNet و Maxpooling را اضافه میکنیم و در آخر با استفاده از تابع () Flatten تمامی فیچرها را در یک تنسور یک بعدی قرار داده و بعد به دو شبکه عصبی تک لایه با 128 نورون وا اکتیویشن فانکشن relu و 47 نورون و اکتیویشن فانکشن softmax میسپاریم و خروجی و دریافت میکنیم که دیتکت میکنه عکس ما کدوم لیبل را دارد . در نمودار مقدار تغییرات loss که کاهشی است و دقت ترین و ولیذیشن را که افزایشی است را در هر epoch میتوان مشاهده کرد . لازم به ذکر است که ما یک callback فانکشنی و را در زمان ترین مدل تعریف کردیم که مقدار درصد ترین مدل و بتونه کنترل کنه که ما تا 0.92 درصد تریشلد قرار دادیم . در آخر در زمان تست مقدار دقت مدلمون 0.86 میباشد .





استفاده از الگوریتم k تا از نزدیک ترین همسایگی (Bruteforce)

الگوریتم K Nearest Neighbor در گروه یادگیری تحت نظارت قرار می گیرد و برای طبقه بندی (رایج ترین) و رگرسیون استفاده می شود. این یک الگوریتم همه کاره است همچنین برای محاسبه مقادیر از دست رفته و نمونه گیری مجدد مجموعه داده ها استفاده می شود. همانطور که از نام (K Nearest Neighbor) پیداست، Kنزدیکترین همسایه ها (نقاط داده) را برای پیش بینی کلاس یا مقدار پیوسته برای Datapoint (داده جدید) جدید در نظر می گیرد. پیچیدگی محاسباتی (Bruteforce) الگوریتم KNN عمدتاً به اندازه مجموعه داده، تعداد ویژگی ها و مقدار k بستگی دارد. پیچیدگی زمانی الگوریتم KNN برای یک نقطه O(nd) است که n تعداد نمونه های آموزشی و d تعداد ویژگی ها است. این به این دلیل است که برای هر نقطه ، الگوریتم باید فاصله بین نقطه و هر نقطه دیگر در مجموعه داده را محاسبه کند. با افزایش تعداد ویژگی ها یا اندازه مجموعه داده، پیچیدگی محاسباتی KNN نیز به طور قابل توجهی افزایش می یابد، که آن را از نظر محاسباتی گران و برای مجموعه داده های بزرگ غیر عملی می کند. در الگوریتم زیر ما برای محاسبه مقدار فاصله نقطه جدید از نقاط همسایه از روش های زیر میتوان استفاده کرد :

Euclidean Distance : فاصله‌ی اقلیدسی برای محاسبه‌ی فاصله‌ی میان دو نقطه در یک صفحه یا یک فضای سه‌بعدی استفاده می‌شود. فاصله‌ی اقلیدسی براساس قضیه‌ی فیثاغورس است .که فرمول آن برابر است با :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Manhattan Distance : اگر به جای مربع فاصله بین مولفه‌ها، از قدر مطلق فاصله بین مولفه‌های نقاط استفاده شود، تابع فاصله را "منهتن" (Manhatan) می‌نامند. اگر x و y دو نقطه با p مولفه p بعدی باشند، شیوه محاسبه فاصله منهتن به صورت زیر خواهد بود:

$$D_{man} = \sum_{i=1}^{p} |x_i - y_i|$$

Minkowski Distance : حالت کلی‌تری از فاصله اقلدیسی و منهتن را برای اشکال محدب یا کوژ میباشد . اگر A و B دو نقطه در فضای p بعدی باشند، فاصله مینکوفسکی برایشان به صورت زیر محاسبه می‌شود. پارامتر فاصله مینکوفسکی در اینجا d در نظر گرفته شده است .

$$D_{mink}(A, B; d) = (\sum_{i=1}^{p} |x_i - y_i|^d)^{\frac{1}{d}}$$

که ما چون با تصاویر که دو بعدی هستند سر و کار داریم از روش مینکفسکی استفاده کردیم .

و با نوشتن تابع هایی برای فیت کردن مدل و محاسبه زمان آن و یافتن مقدار دقت مدل با استفاده از داده ارزیابی که برابر با 0.77 در صد بود به به دقت نسبتا خوبی رسیدیم که با تنظیم کردن پارامترها ی الگوریتم که شامل نوع فاصله و تعداد نزدیکترین همسایگی و امتحانشون روی داد های تست به نتیجه زیر میرسیم :

```
The best model is the 7 th model. Accuracy: 0.7651595744680851
Info
{'algorithm': 'brute', 'leaf_size': 30, 'metric': 'minkowski', 'metric_params': None, 'n_jobs': None, 'n_neighbors': 20, 'p': 2, 'weights': 'distance'}
```

با استفاده از confution matrix به درستی پیش بینی مقادیر میتوان پی برد برای تمامی لیبل ها :

مقایسه الگوریتم ها و نیجه گیری

با مقایسه الگوریتم ها میتوان به نتایج زیر دست یافت :

1- مقدار ران تایم الگوریتم KNN نسبت به CNN کمتر بوده .
2- دقت الگوریتم CNN نسبت به KNN بالاتر است .
3- پیچیدگی زمانی الگوریتم KNN کمتر است .
4- امکان افزایش مقدار تظیم دقت مدل با استفاده از تابع callback وجود دارد .

پایان

منابع

1– Vega-Rodríguez, M.A. Review: Feature Extraction and Image Processing. Comput. J. 2004, 47, 271–272.

2- Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. 2015, 7, 14680–14707.

3- Lu, X.; Sun, H.; Zheng, X. A feature aggregation convolutional neural network for remote sensing scene classification. IEEE Trans. Geosci. Remote Sens. 2019, 57, 7894–7906

4- Minetto, R.; Segundo, M.P.; Sarkar, S. Hydra: An ensemble of convolutional neural networks for geospatial land classification. IEEE Trans. Geosci. Remote Sens. 2019, 57, 6530–6541.

5- Cheng, G.; Yang, C.; Yao, X.; Guo, L.; Han, J. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs. IEEE Trans. Geosci. Remote Sens. 2018, 56, 2811–2821.

6- Sun, H.; Li, S.; Zheng, X.; Lu, X. Remote sensing scene classification by gated bidirectional network. IEEE Trans. Geosci. Remote Sens. 2019, 58, 82–96.

7- He, N.; Fang, L.; Li, S.; Plaza, J.; Plaza, A. Skip-connected covariance network for remote sensing scene classification. IEEE Trans. Neural Netw. Learn. Syst. 2019, 31, 1461–1474.

8- Chen, G.; Zhang, X.; Tan, X.; Cheng, Y.; Dai, F.; Zhu, K.; Gong, Y.; Wang, Q. Training small networks for scene classification of remote sensing images via knowledge distillation. Remote Sens. 2018, 10, 719.

9- Zhang, F.; Du, B.; Zhang, L. Scene classification via a gradient boosting random convolutional network framework. IEEE Trans. Geosci. Remote Sens. 2015, 54, 1793–1802

10 - Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G. Recent Advances in Convolutional Neural Networks. Pattern Recognit. 2018, 77, 354–377

11- Nair, V.; Hinton, G. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair. 2010, Volme 27, pp. 807–814. Available online: https://dl.acm.org/doi/10.5555/3104322.3104425 (accessed on 1 June 2021).

12- Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. 2013. Available online: https://www.mendeley.com/catalogue/a4a3dd28-b56b-3e0c-ac53-2817625a2215/ (accessed on 1 June 2021).

13- He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. IEEE Int. Conf. Comput. Vis. (ICCV 2015) 2015, 1502, 1026–1034.

14- Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. arXiv 2015, arXiv:1505.00853.

15- Zeiler, M.; Fergus, R. Visualizing and Understanding Convolutional Neural Networks. In European Conference on Computer Vision; Springer: Cham, Switzerland, 2013; Volume 8689.

16- Sainath, T.N.; Mohamed, A.r.; Kingsbury, B.; Ramabhadran, B. Deep convolutional neural networks for LVCSR. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8614–8618.