

## تفاوت بین لایه های fully connected, locally connected و convolutional

در ابتدا تفاوت میان fully connected و convolutional را توضیح میدهم و در نهایت locally.

لایه های کانولوشنی بسیار اختصاصی تر و کارآمد تر از لایه های فولی هستند. در لایه های فولی هر نورون به تمام نورون های لایه قبل خود متصل میباشد و هر نورون وزن های خود را دارد. یک الگوی اتصال با هدف عمومی دارد و برای مشخصه های داده فرضی وجود ندارد. این نوع اتصال مموری زیادی استفاده میکند و بار محاسباتی دارد. در مقابل در لایه های کانولوشنی هر نورون به تعداد کمی نورون های نزدیک بهم (به صورت محلی) در لایه قبل خود متصل میباشد و مجموعه وزن ها برای تمام نورون ها مشترک میباشد. این نوع اتصال میتواند به صورت کانوال کردن یک مجموعه از فیلتر ها با داده یا تصاویر تفسیر شوند. مقادیر این فیلتر ها همان وزن های ما هستند. این نوع اتصال زمانی منطقی بنظر میرسد که ویژگی ها وابستگی مکانی دارند مثلاً رنگ یک پیکسل باید در کنار پیکسل های همسایه اش معنا شود. استفاده رایج از لایه های کانولوشنی برای داده های تصویر میباشد.

لایه های لوکالی بسیار شبیه به لایه های کانولوشنی هستند با یک تفاوت مهم. در لایه کانولوشنی فیلتر برای تمام نورون های خروجی مشترک است. به عبارت دیگر ما از یک فیلتر برای محاسبات تمام نورون های خروجی استفاده میکنیم درحالی که در لایه های لوکالی هر نورون فیلتر خود را دارد. یعنی لوکالی از ویژگی shared weights لایه های کانولوشنی استفاده نمیکند.

**کاربرد لایه fully connected:** این لایه وابستگی مکانی را لحاظ نمیکند و برای کاربرد تصویر حکم کلسیفایر خطی را دارند و در آخرین لایه ها استفاده میشوند تا مثلاً کاری مانند دسته بندی تصویر را انجام دهند.

**کاربرد لایه locally connected:** به طور ویژه محققان از خصوصیت این لایه های به خصوص در تایید و شناسایی چهره استفاده میکنند. در لایه های آخر (قبل از fully connected) شبکه که زمان تصمیم گیری است استفاده از locally connected ها مفید میباشد چون میخواهیم از هر بخش از تصویر یک مشخصه یاد بگیرد. ضمن اینکه در لایه های آخر تصویر ما کوچک شده است مثلاً از  $200 \times 200$  به  $16 \times 16$  رسیده ایم و پارامتر ها به نسبت کمتر زیاد میشوند.

**کاربرد convolutional:** همانطور که در تفاوت بین لایه ها اشاره شد این لایه ها در داده های تصاویر استفاده میشوند زیرا تصاویر وابستگی مکانی دارند. همچنین زمانی که میخواهیم یک مشخصه ساده و عمومی از تصویر استخراج کنیم و برای ما اهمیت ندارد این مشخصه بالا، پایین، راست و به طور کلی کجای تصویر میباشد از این لایه ها (در ابتدا و اواسط شبکه) استفاده میکنیم.

## مزایا و معایب stride:

مزایا:

Activation map کوچکتر به دست می آید مثلا اگر stride برابر با 2 باشد تعداد واحد های نقشه ویژگی تقسیم بر 4 میشود. در نتیجه تعداد پارامتر ها کمتر و محاسبات سریعتر میشود.

مورد دیگری که استاد در کلاس پردازش تصویر اشاره کردند کاهش overfitting بود. چون کار شبکه برای حفظ داده های train سختر میشود و امیدوار میشویم روی داده های validation دقت بالاتری به دست آوریم.

معایب:

از دست دادن اطلاعات. اگر الگویی در پوزیشنی که از آن رد میشود با شدت بیشتری قابل مشاهده باشد، آن اطلاعات از بین می رود. برای جلوگیری از این اتفاق از لایه های pooling استفاده میشود که کاهش بعد را با روش هوشمندانه تری انجام میدهد و روش آن ادغام پیکسل های همسایه با روش هایی مانند max یا average میباشد.