

Deep learning - theory - Hw3 - 4/20/2018 محمدحسین سلیمانزاده

۱) برای مسئله ۵ کلاس به هر حالی که هر تصویر 10×10 پیکسل را داریم. مسئله کانولوشن می تواند بر این پیکسل (از جنس فیلتر) بهتر استخراج نماید. همچنین فیلترهای کانولوشن نسبت به دیدن به ازای 10×10 تر از یک مسئله فوی کانولوشن است. همچنین لایه های کانولوشن نسبت به Shift در داده ها نیز robust تر هستند. همچنین لایه های کانولوشن نسبت به آنتینیتینگ مقاوم ترند (میرفتن، اسکیل، نورپردی)

۲) * تصویربرداری ۳ کانال دارد تصویربرداری ۱ 2×2 و ۱ 8×8 دارای فیلتر

$$\begin{array}{c} 128 \times 128 \times 14 \\ \hline 128 + 128 = 256 + 1 \\ \text{Weights} \end{array}$$

تعداد فیلترها

بایاس

$$5 \times 5 \times 3 \times 14 + 14$$

* تعداد بایاس ها:

max pooling

$$\left(\frac{128}{2} \times \frac{128}{2} \times 14 \right) = 64 \times 64 \times 16$$

خردی لایه اول:

$$\frac{64}{2} \times \frac{64}{2} \times 16 = 32 \times 32 \times 16$$

خردی لایه دوم:

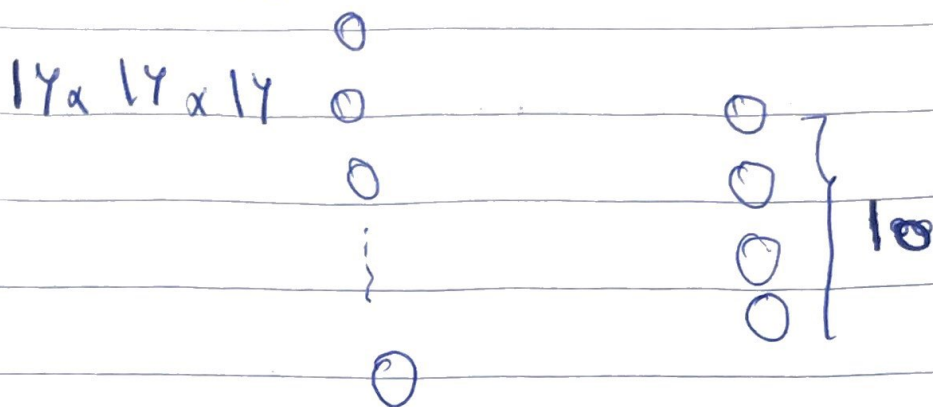
$$\frac{32}{2} \times \frac{32}{2} \times 16 = 16 \times 16 \times 16 = 4096$$

خردی لایه سوم:

$$5 \times 5 \times 3 \times 14 + 14 + 2(5 \times 5 \times 16 \times 16 + 16) = 14032$$

لایه های Max pooling, Relu و بایاس ها

اگر در لایه آخر FC برای اکسل قرار دهیم:



تعداد پارامترهای وزن این لایه: 4096×10 تعداد بایاس ها: 10

تعداد کل پارامترهای شبکه: $55002 = 40970 + 14032$
 قسمت اول: کانولوشن
 قسمت دوم: بایاس ها

* برای هر نورون خروجی به تعداد 4 نورون برای Max pooling احتیاج داریم

که این 4 نورون خروجی 4×4 نورون در لایه قبلی کانولوشن هستند

به همین ترتیب برای لایه دوم از Max pooling 12×12 و بایاس ها 12×12 می باشد

حال برای لایه اول 28×28 (قبل از لایه Max pooling) و قبل از کانولوشن 32×32

است. به عبارتی هر نورون خروجی از 32×32 نورون در ورودی تأثیر می پذیرد

(۳) آید. اصلی مانند شبکه U-Net این است که در انتهای بخش جمع شونده شبکه
 یک سری لایه کانولوشن قرار دهیم تا این تفاوت که به جای استفاده از Pooling از Upsampling
 استفاده می‌کنیم تا به این ترتیب در طول فرایند افزایشی یا به عبارتی حرکت به سمت عقب تر
 context - patch (بازه pixel) دقت بالاتری داریم.

طبیعی است که ما در شبکه فیلترهای کانولوشن را از بالا به پایین پس از Upsample
 کردن آن دوباره در طول فرایند افزایشی می‌گذاریم.

* برای localize به object در Semantic Segmentation فیلترهای کانولوشن بالا در مسیر رفت

با فیلترهای Upsample می‌کنیم. در مسیر برگشت ترکیب می‌شوند که باعث افزایش بهر و عمق
 localization در patch می‌شود.

* در حوزه شبکه یادگیری عمیق در تدریس دنیا Training با مسائل یادگیری ~~مشکلات~~
 دنیا های Invariant مولد هستیم که کمک به تشخیص و حفظ ویژگی‌ها با تغییرات بالا در ورودی
 می‌توانیم با Augment کردن دنیا است آموزش به دست می‌دهیم که به شبکه کانولوشن داده می‌شود
 (در شبکه های Semantic مسائل (classification, localization) هست.

۲-۳ در پایین ترین لایه های دیر ما ۱۶۱ و ۱۶۱ فیلتر خواهد داشت

با توجه به مقادیر ۱۶۰ و ۱۶۰ این شبکه در مقادیر پارامترها می‌تواند

برای لایه ۱۶۱ فیلتر ۳ به ازای هر لایه درون این لایه داریم:

$$3 \times 3 \times 64 \times 161 + 161 + (2 \times 2 \times 161 \times 161 + 161)$$

تفاوت معماری:

DenseNet و ResNet هر دو به لایه‌های بعدی Skip connection دارند. به عبارتی در مجموع L لایه، L Skip Connection است. این در حالی است که در DenseNet هر لایه به تمامی لایه‌های بعد از خود یک کانکشن دارد. به این صورت که ورودی به L لایه و خروجی لایه دوم به $L-1$ و ... دارد پس در مجموع $\frac{L(L+1)}{2}$ کانکشن بین لایه‌ها را داریم.

تفاوت عمده این دو شبکه این است که در DenseNet برخلاف ResNet نقشه ویژگی در هر لایه از جمع نقشه ویژگی لایه قبل و خروجی لایه قبل بدست می‌آید. بلکه که لازم است که هم قراردان آن‌ها (concatenate) بدست می‌آید. این عمل به شبکه اجازه می‌دهد این ویژگی‌ها را دوباره آموزش ببیند و به این ترتیب تعداد پارامترهای آن خیلی کمتر از ResNet می‌شود.

* با بررسی ارتباط بین لایه‌های در شبکه DenseNet هر لایه به طور مستقیم به لایه‌های بعدی وصل می‌گردد (معماری Skip Connection) و به این ترتیب با کاهش Gradient Vanishing مواجه می‌شود.

همان‌طور که در قسمت اول این سوال گفته شد، عمل Concatenate کردن ورودی لایه قبل و خروجی لایه قبل می‌تواند از آموزش مجدد پارامترهای مربوط به نقشه ویژگی ورودی لایه قبل جلوگیری نکند و به این ترتیب حجم محاسباتی را به صورت چشمگیری کاهش می‌دهد.

۳-۴

* بهایات 3 لایه DenseNet

$$x_l = H_l \left(\underbrace{[x_0, x_1, \dots, x_{l-1}]}_{\text{ورودی لایه}} \right)$$

$$\text{input} = [x_0, x_1, x_2]$$

با فرض اینکه لایه اول ۳۲ فیچر دریافتی نماید (۳۲ = ۳۲) ورودی لایه دوم

$$224 = 32 + 44 + 148$$

فیچر

* با توجه به رابطه $K_0 + K(l-1)$ ورودی لایه سوم $32 + 44 \times 2$ کال

ورودی دریافتی شد و آن $K=24$ فیچر مستقیم از لایه قبلی دریافتی نماید.

$$32 + 72 = 104$$