یادگیری عمیق

دکتر فاطمی زاده

مهندسی برق

برنا خدابنده ۴۰۰۱۰۹۸۹۸

تمرین ۳

شبکه‌های عمیق کانولوشنی

۲۴ آذر ۱۴۰۲

یادگیری عمیق

تمرین ۳

برنا خدابنده ۴۰۰۱۰۹۸۹۸

سوالات نظری

سوال ۱

یک مسئله دسته‌بندی با ۵ دسته، که دیتاست، شامل تصاویری به اندازه 10×10 پیکسل میباشند، داریم. دو شبکه عصبی یک لایه را به صورت زیر در نظر بگیرید. توضیح دهید کدام یک انتخاب بهتری میباشد؟

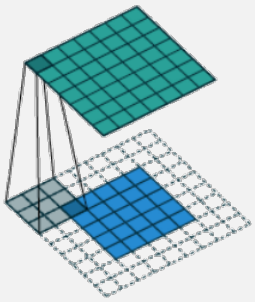
- یک لایه fully connected که ورودی آن، flatten (بردار شده) تصاویر دیتاست میباشد.
- یک لایه کانولوشن که که در آن ۵ فیلتر به اندازه 10×10 داریم.

در این ساختار، این ۲ شبکه با هم معادل اند. زیرا هر کدام از این فیلتر ها، سایزی برابر با سایز عکس ورودی دارند، در نتیجه هر کدام از این فیلتر ها به مانند یک perceptron عمل میکنند.
در نتیجه گویا ۵ perceptron به صورت fully connected بین عکس و خروجی داریم که به طور کامل معادل شبکه fully connected عمل میکند.
ولی به طور کلی شبکه های کانولوشنی برای این نوع استفاده ها بهتر عمل میکنند، ولی در این ساختار و با این ابعاد ورودی و فیلتر، معادل شده با یک شبکه fully connected میشوند.

سوال ۲

فرض کنید دیتاستی داریم شامل تصاویر رنگی به اندازه ۱۲۸ × ۱۲۸۰ میخواهیم یک شبکه عصبی کانولوشن برای آن طراحی کنیم.

- اندازه خروجی و تعداد پارامترهای لایه اول کانولوشن را محاسبه کنید اگر ۱۶ فیلتر ۵× ۵ با stride = ۱ و padding = ۲ داشته باشیم.



عکس با استفاده از فیلتر 3×3 است و ورودی های 5×5 ولی برای شهود کمک کننده است.

$$\begin{aligned}\text{parameters} &= 16 \times 5 \times 5 + 16 = 416 \\ \text{output size} &= 16 \times (128 + 2 \cdot 2 - (5-1))^2 \\ &= 16 \times 128 \times 128 = 262,144\end{aligned}$$

- فرض کنید هر لایه، ۳ قسمت شامل : کانولوشن، max pooling و تابع فعالسازی (ReLU) را دارا می باشد، که لایه های کانولوشنی، هر کدام شامل ۱۶ فیلتر ۵× ۵ با stride = ۱ و padding = ۲ و لایه‌های max pooling همگی ۲ × ۲ با stride = ۲ هستند. ۳ لایه با این مشخصات را پشت سر هم در نظر بگیرید. اندازه تنسور در لایه خروجی نهایی و تعداد پارامترهای این ۳ لایه را حساب کنید.

همانطور که مشاهده شد، لایه های کانولوشن با سایز 5×5 با $padding = 2$ به صورت same size کار میکنند، حال کافیست لایه های max-pooling را در نظر بگیریم.
لایه های max-pooling با 2×2 ,stride=2 به این صورت اند که ابعاد ورودی را نصف میکنند. پس سایز هر لایه به صورت زیر خواهد بود.

$$\begin{aligned}16 \times 128 \times 128 &\xrightarrow{\text{conv}} 16 \times 128 \times 128 \xrightarrow{\text{pooling}} 16 \times 64 \times 64 \\ &\xrightarrow{\text{conv}} 16 \times 64 \times 64 \xrightarrow{\text{pooling}} 16 \times 32 \times 32 \\ &\xrightarrow{\text{conv}} 16 \times 32 \times 32 \xrightarrow{\text{pooling}} 16 \times 16 \times 16\end{aligned}$$

پس خروجی سایز $16 \times 16 \times 16$ را دارد.
برای تعداد پارامتر نیز، تعداد پارامتر فیلتر ها را باید شمرد، که هر لایه کانولوشنی تعداد $16 \times 5 \times 5$ پارامتر دارد.

$$\text{parameters} = 3 \times 16 \times 5 \times 5 + 16 = 3 \times 400 + 16 = 1216$$

- فرض کنید هدف، حل یک مسئله ،classification که شامل ۱۰ دسته هست،می باشد.تعداد کل پارامترهای شبکه را در این حالت حساب کنید.

ساده ترین کاری که میتوان کرد، این است که صرفا خروجی فیچر های بدست آمده از سوی لایه های کانولوشنی را با استفاده از یک لایه خطی، به حالت طبقه بندی با softMax در آوریم. در این صورت:

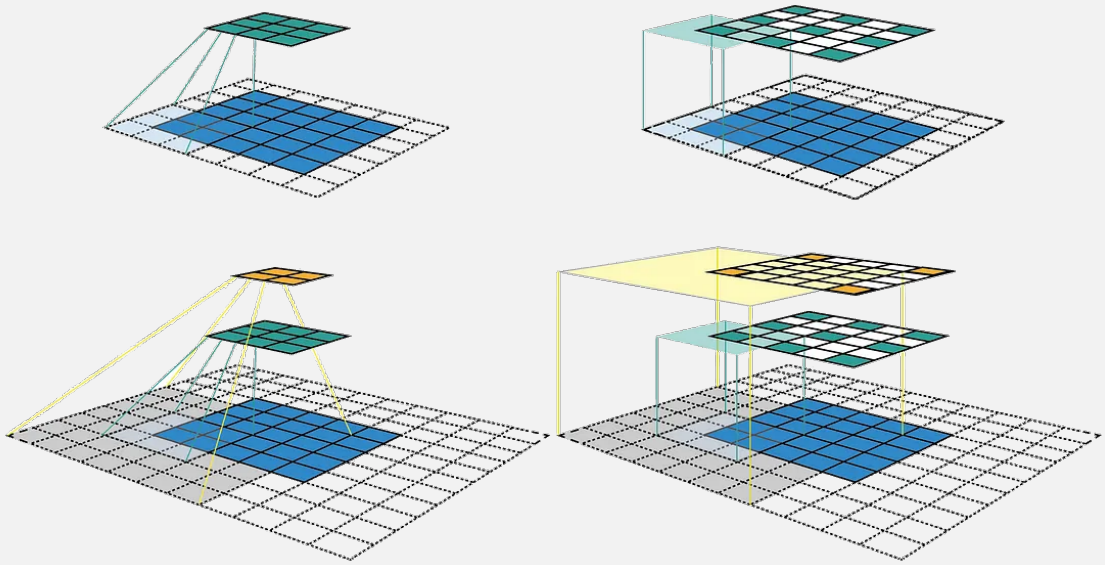
$$\text{parameters classifier} = 16 \times 16 \times 16 \times 10 + 10 = 40970$$

که در این صورت کل پارامتر ها مقدار ۴۲۱۷۰ پارامتر خواهد شد، برای کاهش پارامتر، میتوانیم در این لایه آخر نیز pooling انجام دهیم. برای مثال اکر از Global Average pooling در سطح کانال استفاده کنیم، تغداد پارامتر به صورت زیر خواهد بود

$$\text{parameters classifier}_{GAP} = 16 \times 10 + 10 = 170$$

که تعداد کل پارامتر را به ۱۳۷۰ تقلیل میدهد، که البته احتمالا برای این مسئله کم باشد.

- در این قسمت، با یک مفهوم مهم آشنا می شویم. Receptive field بیانگر این است که نورون خروجی، تحت تاثیر چه مقدار از نورون های ورودی می باشد. در حقیقت تعیین می کند هر نورون خروجی از چه ناحیه‌ای با چه اندازه‌ای از تنسور ورودی تاثیر می پذیرد. حال Receptive field را برای یک نورون خروجی لایه سوم (قبل از لایه fully connected) بررسی کنید. (برای فهم بهتر این مفهوم میتوانید به این لینک مراجعه کنید)



طبق اینکه داریم $s = 1$ برای تمام لایه ها، در هر لایه خواهیم داشت:

$$r_{out} = r_{in} + (k - 1), \quad k : \text{size kernel}$$

در نتیجه برای receptive field لایه سوم خواهیم داشت:

$$r_3 = 1 + 3 \times (5 - 1) = 1 + 12 = 13$$

پس یک قسمت 13×13 از عکس اصلی را میبیند.

سوال ۳

در این تمرین قصد داریم به بررسی دو شبکه‌ی معروف یعنی Densely Connected Convolutional Networks [۱] و U-Net[۲] بپردازیم. برای بررسی هرچه بهتر این دو شبکه بهتر است به لینک های زیر مراجعه نمایید.

۱. **سوالات مفهومی مربوط به U-Net :**

- ویژگی اصلی شبکه‌ی U-Net که آن را از یک شبکه کانولوشنی عادی متمایز می دارد چه می باشد و دلیل اینکه ما شاهد یک ساختار U شکل هستیم چه می باشد؟

ساختار U شکل این شبکه اصلی ترین عامل تمایز بین این شبکه و شبکه های کانولوشنی معمولی است، این ساختار شامل ۳ قسمت اصلی و skip connection بین آنها است که دقت این شبکه را برای کار های segmentation بسیار بالا میبرد.

- **مسیر با کاهش بعد (Encoder)** : شامل چندین لایه کانولوشنی با پولینگ که مهم ترین feature ها را استخراج میکنند.
- **Bottleneck** : قسمت میانی و نازک که دارای مهم ترین feature ها یاد گرفته شده توسط Encoder است.
- **مسیر با افزایش بعد (Decoder)** : این قسمت با استفاده از داده های skip connection و چندین لایه upconv و up-sampling تصویر را بزرگ کرده تا segmentation map نهایی را بسازد.

حالت U شکل این ساختار کمک میکند تا اطلاعات لوکال و گلوبال را حفظ کنیم.

- می دانیم که Skip connection ها نقشی پررنگ در این شبکه‌ها دارند، دلیل حضور این مورد را در شبکه‌ی U-Net بیان کنید.

این اتصالات به ما کمک میکند تا اطلاعات مکانی و دیگر اطلاعاتی که ممکن است در لایه های down-sampling و encode کردن از دست رفته باشد را باز بدست آوریم، پس به ساختار کمک میکند تا اطلاعات high-level و low-level را با هم ادقام کنیم تا کار دقیق تری انجام دهیم.

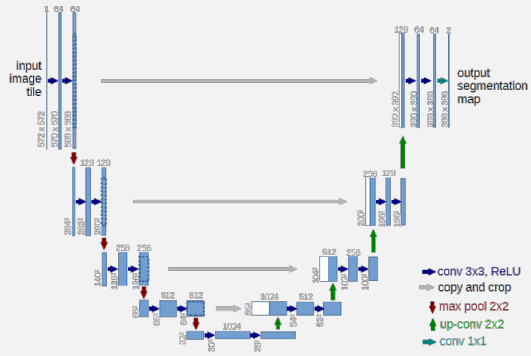
- چرا این نوع از اتصالات در تصاویر پزشکی دارای اهمیت بیشتر می باشند و چه کمکی به ما در دامنه‌ی تشخیص موارد پزشکی می کنند؟

این شبکه به طور خاص برای عملیات segmentation طراحی شده است، اتصالات خاص این نوع معماری همراه با skip connection ها به ساختار کمک میکند که برای مثال در تشخیص تومور، با ادقام اطلاعات سطح بالا و اطلاعات مکانی و سطح پایین، مرز تومور ها را به دقت تشخیص دهند و چیز های غیرمعمول لوکال را تشخیص دهند.
دقت بالای این سیستم نیز آن را کاندید خوبی برای کار های پزشکی میکند.

۲. سوالات محاسباتی مربوط به U-Net :

- تصور کنید که ابعاد تصویر ورودی ما برای این شبکه ۲۵۶ × ۲۵۶ می باشد. حال فرض می شود که در این معماری هرلایه در انکدر ابعاد را به نصف کاهش می دهد و در دیکدر دو برابر می کند. در پایین ترین لایه (عمیق ترین لایه) این معماری، فضای ویژگی ما چند پیکسل خواهد داشت؟

در ساختار U-Net دارای ۴ لایه انکدر هستیم، در نتیجه:



$$256^2 \to 128^2 \to 64^2 \to 32^2 \to 16^2$$

ولی با تعداد 1024 کانال اطلاعات خواهیم بود، با در نظر گرفتن کاهش ابعاد ناشی از کانولوشن ولی:

$$256^2 \to 252^2 \to 126^2 \to 122^2 \to 61^2 \to 57^2 \to 28^2 \to 24^2 \to 12^2 \to 8^2$$

که برای سایز خروجی باید در واقع این عدد را ضرب در $c \times c$ نیز کرد، که برای لایه آخر $c = 1024$.

- در U-Net، فرض کنید انکودر دارای لایه‌هایی با ۶۴، ۱۲۸، ۲۵۶ و ۵۱۲ فیلتر است. اگر هر لایه کانولوشن از کرنال های ۳ × ۳ استفاده بکند، تعداد پارامترهای لایه کانولوشن دوم انکدر را محاسبه کنید.

لایه دوم شامل دو لایه کانولوشنی با فیلتر های یکسان و یک لایه پولینگ است، لایه پولینگ پارامتر قابل یادگیری ندارد، ولی لایه های کانولوشنی هر کدام شامل ۱۲۸ فیلتر 3×3 میباشند، به تعداد کانال نیز باید دقت شود، که تعداد پارامتر در نتیجه:

$$\text{parameters} = 128 \times (3 \times 3) \times 64 + 128 + 128 \times (3 \times 3) \times 128 + 128 = 221,440$$

۳. سوالات مفهوم DenseNet :

- تفاوت های اصلی بین ResNet's residual connections و DenseNet's Dense connections را بیان کنید. درمورد هرکدام از موارد گفته نیز، توضیح مختصری بدهید.

در ResNet ما از Skip connection استفاده میکنیم که یک یا چند لایه مشخص را رد میکنند، به صورتی که انگار تابع نهایی به صورت $H(x) = F(x) + x$ است، و ما قسمت غیرخطی را محاسبه میکنیم، در DenseNet ما از Dense connection استفاده میکنیم، به ایت صورت که هر لایه، خروجی تمامی لایه های قبل خود را دریافت میکند که انگار به صورت $H(x) = H_l([x_0, x_1, \ldots, x_{l-1}])$ میباشد.

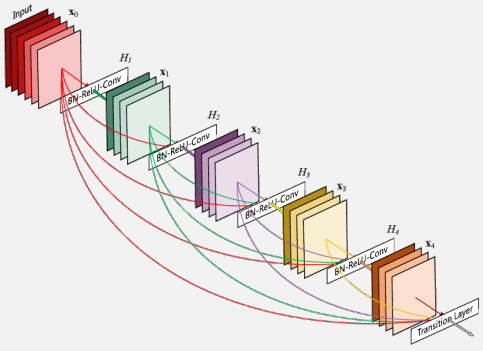
- **ResNet** : در این ساختار ما از چندین Residual block استفاده میکنیم که هماطور که گفته شد دارای skip connection هستند، این نوع اتصالات هم به generalization کمک کرده هم به شارش گرادیان کمک میکند.
- **DenseNet** : این ساختار از چندین Dense block استفاده میکند که با دادن اطلاعات تمامی لایه های پیشین به ورودی هر لایه، ساختار را تشویق به استفاده دوباره و چند باره از فیچر ها، بهبود انتقال اطلاعات و بهبود شارش گرادیان در backpropagation میکند.

- بیان کنید که DenseNet چگونه مشکل vanishing gradient را کاهش میدهد و مزیت آن چه می باشد؟

ساختار DenseNet مشکل گرادیان را اینگونه بهتر میکند که با استفاده از Dense connection، مسیر های کوتاه تری برای گرادیان وجود دارد که backpropagation را انجام دهد، و در عمل انگار خروجی هر لایه و پارامتر های هر لایه یک مسیر مستقیم تا اطلاعات loss function دارند که به یادگیری بهتر جفت فیچر های low-level و high-level کمک میکند.
مزیت های دیگر این است که طبق اینکه سیستم به اطلاعات لایه های قبل دسترسی دارد و آنها را دوباره استفاده میکند، ساختار به این تشویق میشود که فیچر های جدید تر و مفید تری یاد بگیرد بجای یادگیری فیچر های تکراری و بی استفاده.
همچنین این استفاده دوباره از فیچر ها از کم شدن دقت با افزایش زیاد عمق که ساختار های معمولی دچار آن هستند جلوگیری میکند و از اشباع شدن سیستم نیز جلوگیری میکند.

۴. سوالات محاسبه‌ای DenseNet :

- در یک DenseNet با سه لایه در یک Dense Block اگر لایه اول ۶۴ فیچر مپ تولید کند، لایه دوم ۱۲۸ فیچر مپ و لایه سوم ۲۵۶ فیچر مپ تولید کند، لایه سوم چند فیچر مپ ورودی را دریافت خواهد کرد؟



یک لایه در DenseNet در واقع concatenate شده خروجی تمامی لایه های قبل خود را میگیرد، در نتیجه میتوان به صورت زیر نوشت.

$$x_3 = H_3([x_1, x_2])$$
$$|[x_1, x_2]| = |x_1| + |x_2| = 64 + 128 = 192$$

در نتیجه ۱۹۲ فیچر مپ ورودی داریم.

- با در نظر گرفتن نرخ رشد k در DenseNet، اگر هر لایه k فیچر مپ جدید تولید کند و ورودی یک dense block دارای ۳۲ کانال باشد، اگر k = ۲۴ باشد لایه سوم در بلوک چند کانال خروجی خواهد داشت؟

در شبکه DenseNet پارامتر k توصیف کننده تعداد فیچر مپی است که هر لایه به ورودی لایه بعد اضافه میکند، محاسبه به صورت زیر است.

$$C_{out} = \overbrace{\underbrace{C_{in}}_{\text{input}} + k}^{\text{first layer}} \underbrace{+k}_{\text{second layer}} + k = C_{in} + k \times n \Rightarrow 32 + 72 = 104$$

در نتیجه ۱۰۴ کانال خروجی خواهیم داشت.

*

References

[1] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” 2018.

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.