محمدرضا ایرزاده
۴۰۱۲۰۶۹۹

تمارین سری دوم درس Deep Learning

① (ب)

Input	۱۲۸ × ۱۲۸ × ۳	
Conv 9,32	۱۲۰ × ۱۲۰ × ۳۲	$\lfloor \frac{۱۲۸-۹}{۱} + ۱ \rfloor = ۱۲۰$
Maxpool, 2	۶۰ × ۶۰ × ۳۲	$\lfloor \frac{۱۲۰-۲}{۲} + ۱ \rfloor = ۶۰$
Conv 5,64	۵۶ × ۵۶ × ۶۴	$\lfloor \frac{۶۰-۵}{۱} + ۱ \rfloor = ۵۶$
Maxpool 2	۲۸ × ۲۸ × ۶۴	$\lfloor \frac{۵۶-۲}{۲} + ۱ \rfloor = ۲۸$
Conv 5,64	۲۴ × ۲۴ × ۶۴	$\lfloor \frac{۲۸-۵}{۱} + ۱ \rfloor = ۲۴$
Maxpool 2	۱۲ × ۱۲ × ۶۴	$\lfloor \frac{۲۴-۲}{۲} + ۱ \rfloor = ۱۲$
flatten	۹۲۱۶ × ۱	$۱۲ \times ۱۲ \times ۶۴ = ۹۲۱۶$
FC-3	۳ × ۱	تعداد وزن‌ها $= ۳۲\left(۹ \times ۹ \times ۳\right) + ۶۴\left(۵ \times ۵ \times ۳۲\right) + ۶۴\left(۵ \times ۵ \times ۶۴\right) + ۹۲۱۶ \times ۳ =$ ۱۸۹۰۲۴

تعداد بایاس‌ها $= ۳۲ + ۶۴ + ۶۴ + ۳ = ۱۶۳$

ب) قدرت توابع فعال‌ساز غیرخطی باعث می‌شود که مدل توانایی یادگیری روابط غیرخطی را نیز داشته باشد. اگر از توابع فعال‌ساز خطی استفاده کنیم، هرچقدر هم که شبکه عمیق باشد، باز هم یک مدل خطی محسوب می‌شود و نمی‌تواند روابط غیرخطی را کشف کند. بنابراین به‌دلیل این قدرت یادگیری شبکه‌های عصبی، از توابع فعال‌ساز غیرخطی استفاده می‌کنیم.

ج) مزایا: از آنجایی که با استفاده از Pooling ابعاد ورودی کمتر می‌شود، بنابراین می‌توان خطر overfit شدن مدل را کاهش داد. از طرف دیگر از آنجایی که پارامترهای یادگیری را کاهش می‌دهد بنابراین هزینه محاسباتی مدل را نیز کاهش می‌دهد.

معایب: استفاده از Max pooling باعث از دست رفتن اطلاعاتی می‌شود که ممکن است مفید باشند. زیرا از بین پیکسل‌های ورودی فقط با شیم آن‌ها را نگه داشته و بقیه پیکسل‌ها را دور می‌اندازد.

د) از آنجایی که در آخرین لایه مدل دسته‌بندی داریم ۳ دسته است بنابراین باید از تابع فعال‌ساز softmax برای استخراج احتمال هریک از دسته‌ها استفاده کرد.

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{۳} e^{z_j}}$$

$$z = [۰.۱, ۰.۵, ۰.۳]^T \Rightarrow softmax(z) = \begin{bmatrix} \frac{e^{۰.۱}}{e^{۰.۱}+e^{۰.۵}+e^{۰.۳}} \\ \frac{e^{۰.۵}}{e^{۰.۱}+e^{۰.۵}+e^{۰.۳}} \\ \frac{e^{۰.۳}}{e^{۰.۱}+e^{۰.۵}+e^{۰.۳}} \end{bmatrix} = \begin{bmatrix} ۰.۲۷ \\ ۰.۴۰ \\ ۰.۳۳ \end{bmatrix}$$

(۲) الف) از آنجایی که فیلترهای لایه‌های کانولوشنی به صورت تصادفی به تصویر ورودی اعمال می‌شود و این فیلترها دارای پارامترهای مشخص و یکسانی است می‌توان گفت که ویژگی weight sharing وجود دارد. این ویژگی باعث می‌شود که پیدا کردن یک الگوی مشخص بالاستفاده از این فیلترها نسبت به ناحیه خاص نباشد و اگر آن الگو در ناحیه‌ی دیگری از تصویر نیز باشد، شبکه با همان فیلتر آن را تشخیص دارد. به این ویژگی لایه‌های کانولوشنی Translational Equivariance گفته می‌گوید تابع $f(x)$ نسبت به تابع $g(x)$ equivariant است اگر $f(g(x)) = g(f(x))$ در اینجا می‌توان گفت $f(x)$ تابع کانولوشن و $g(x)$ تابع translation است.

ب) برای اینکه به یک تفسیر پذیری درست و مشخص برسیم که مدل ما به کدام قسمت‌ها از تصویر ورودی حساسیت دارد و آن قسمت‌ها را ملاک تشخیص قرار می‌دهد نیاز است که گرادیان‌های لایه آخر (بعد از Softmax) را به عقب برگردانیم و به کمک این گرادیان‌ها حاصل می‌شود که کدام نواحی ورودی تأثیر بیشتری در استنتاج دارند. حال در شبکه‌های Deconvolutional گفته شده است که به جای گرادیان ما به این صورت است که آن پارامترهایی که در مرحله forward صفر بودند و همچنین آن پارامترهایی که در مرحله انتشار به عقب منفی هستند را صفر می‌کنیم و تأثیر این دو را به ما را نادیده می‌گیرد. به این روش Guided Backpropagation گفته می‌شود.

روابطی که استفاده می‌شود به صورت زیر است:

activation: $f_i^{l+1} = Relu(f_i^l)$

guided backprop: $R_i^l = (f_i^l > 0) \cdot (R_i^{l+1} > 0) \cdot R_i^{l+1}$

در شبکه‌ی Up convolution برخلاف سایر شبکه‌ها که برای تفسیر پذیری گرادیان‌ها را به عقب برمی‌گرداندند، در اینجا می‌خواهیم که یک شبکه‌ی جدید آموزش داده شود به طوری که با استفاده از این ویژگی‌های حاصل شده بتواند تصویر ورودی را تخمین بزند. در واقع یک شبکه جدید می‌سازیم که ورودی آن ویژگی‌های استخراج شده (ϕ_i) است و خروجی آن تخمین تصویر ورودی (X_i) است. پارامترهای این شبکه نیز با استفاده از مینیمم کردن تابع هزینه به صورت زیر حاصل می‌شود:

$$\hat{w} = \arg\min_w \sum_i \|X_i - f(\phi_i, w)\|_2^2$$

۴) الف) object detector های تک مرحله‌ای، تصویر ورودی را ناحیه‌بندی نمی‌کنند و فقط بر اساس تصویر ورودی یک تعداد ثابت prediction انجام می‌دهند.

object detector های دو مرحله‌ای ابتدا تصویر ورودی را ناحیه‌بندی می‌کنند و تصاویر درون هر یک از نواحی را پیدا می‌کنند سپس در مرحله بعدی، یافته‌های مرحله قبل را بهبود می‌دهند و prediction نهایی را انجام می‌دهند.

ب) تشخیص دهنده Yolo فقط یک بار تصویر ورودی را بررسی می‌کند و آن را به شبکه خود وارد می‌کند و آنرا تشخیص می‌دهد. به همین دلیل نیز به آن تشخیص دهنده ما بروت بینری دارد.

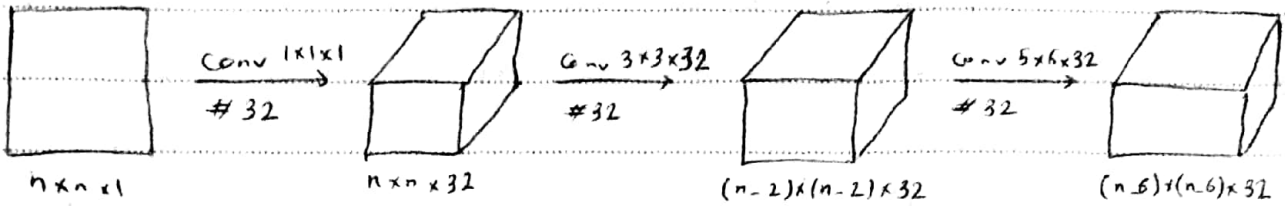
روش RCNN ابتدا نواحی که محتمل حضور اشیا هستند را تابع از دسته‌بندی آن‌ها مشخص میکند. سپس در مرحله بعدی با استفاده از یک شبکه کاملا" کانولوشنی، ویژگی های هر یک از نواحی مرحله قبل را استخراج می‌کند و در نهایت با استفاده از یک دسته‌بند، دسته مربوط به هر یک از نواحی را predict میکند.

۲) RetinaNet یک object detector تک مرحله‌ای است که با استفاده از دو خاصیت feature pyramid و focal loss بهبود یافته است. ویژگی FPN برای این صورت است که در مرحله پایین به بالا به ویژگی‌های با رزولوشن کمتری رسیم و حال با استفاده از یک روی اتصالات کناری و یک شبکه بالا به پایین، ویژگی های لایه‌های جلوی را با ویژگی‌های لایه‌های عمیق ترکیب می‌کند و به این صورت به یک سری ویژگی‌های جدید می‌رسد و سپس این ویژگی‌های جدید را استفاده میکند. در واقع نقش FPN، feature extractor است که به صورت هرمی ویژگی های لایه‌های جلوی را با عمیق ترکیب می‌کند.

---

۳) آ) شبکه Inception از تعدادی از بلوک‌های Inception تشکیل شده است که در هر یک از این بلوک‌ها از فیلترهای با سایزهای مختلف استفاده می‌شود و در نهایت خروجی اعمال این فیلترها با هم concat می‌شوند. البته خروجی این فیلترها باید به گونه‌ای باشد که ابعاد خروجی با ورودی برابر باشد یعنی same convolution که این امر با استفاده از padding روی به درستی قابل انجام است.

هدف از اعمال فیلترهای با سایزهای مختلف در هر لایه می‌تواند این باشد که شبکه را تشویق کند که الگوهای نمای مختلف را در مقیاس‌های scale های متفاوت را بتواند یاد بگیرد و به این طریق دقت شبکه را افزایش دهد.



تعداد پارامترها به صورت زیر محاسبه می‌شود:

~~1×1×1 + 3×3×32 + 5×5×32 + 32 = 1162~~

$$32\left[(1\times1\times1)+1\right] + 32\left[(3\times3\times32)+1\right] + 32\left[(5\times5\times32)+1\right] = 32426$$

در شکل دوم که فیلترها همگی به ورودی که یک کانال دارد اعمال می‌شوند، تعداد پارامترها به صورت زیر محاسبه می‌شود:

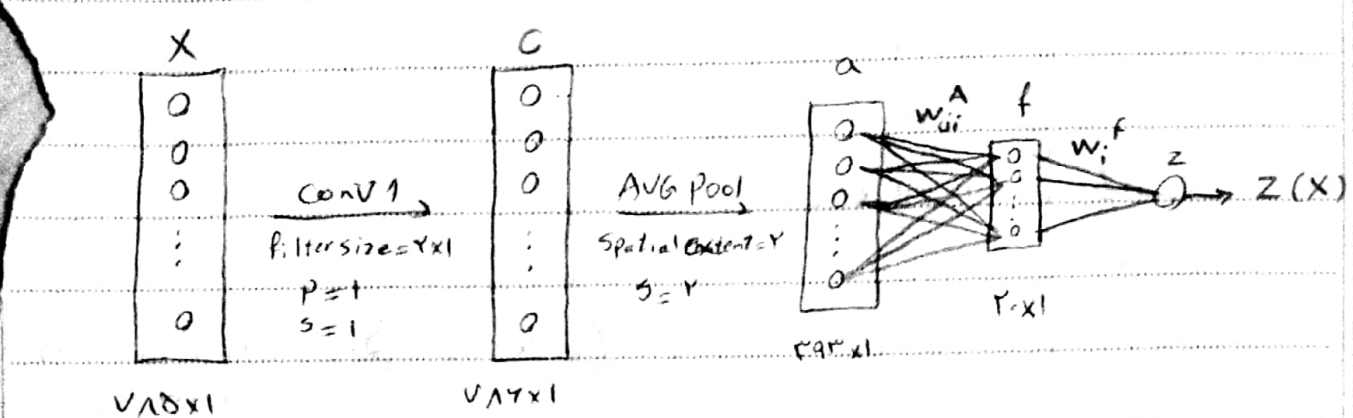
$$32\left[(1\times1\times1)+1\right] + 32\left[(3\times3\times1)+1\right] + 32\left[(5\times5\times1)+1\right] = 1216$$

ب) وقتی که از توابع فعال ساز مانند sigmoid استفاده کنیم، خروجی مقدار در هر لایه عددی بین صفر و یک است. زمانی که در روند Backprop می‌خواهیم با استفاده از Chain Rule گرادیان‌ها را به عقب برگردانیم، این اعداد بین صفر و یک در هم ضرب می‌شوند و سرانجام به صورت عدد کوچک می‌شوند. سرانجام اگر تعداد لایه‌ها خیلی زیاد باشد، گرادیان ما به قسمی گشت به عقب به صفر میل می‌کند و این باعث می‌شود که نتوان پارامترهای مربوط را به صورت مناسب به روزرسانی کرد.

شبکه‌ی Res Net با استفاده از این Skip Connection که مقادیر خروجی لایه‌ی l را به لایه‌ی $l+1$ انتقال می‌دهد می‌تواند اثر محو شدگی گرادیان را کاهش دهد و امکان آموزش شبکه‌های عمیق‌تر را فراهم می‌آورد. در روند Backprop نیز گرادیان در لایه‌ی $l+1$ به لایه‌ی l منتقل می‌شود و به این صورت از محو شدن گرادیان کمتر می‌شود.

۵ الف) تعداد پارامترهایی که شبکه نیاز به یادگیری دارد برابر است با:

$$\underbrace{۲}_{W_c\text{ پارامترهای}} + \underbrace{۱}_{\text{بایاس فیلتر}} + \underbrace{(۴۹۴ \times ۲۰)}_{\text{پارامترهای لایه FC}} + \underbrace{۲۰}_{\text{بایاس لایه FC}} + \underbrace{۲۰}_{\text{پارامترهای لایه Z}} + \underbrace{۱}_{\text{بایاس Z}} = ۷۹۰۴$$



ب) تابع loss برای یک نمونه آموزش برابر است با:

$$cost = -y \lg(z(x)) - (1-y)\lg(1-z(x))$$

$$\frac{\partial cost}{\partial w_{ji}^A} = \frac{\partial cost}{\partial out_{f_i}} \frac{\partial out_{f_i}}{\partial in_{f_i}} \frac{\partial in_{f_i}}{\partial w_{ji}}$$

$$\frac{\partial cost}{\partial out_{f_i}} = \frac{\partial cost}{\partial out_2} \frac{\partial out_2}{\partial in_2} \frac{\partial in_2}{\partial out_{f_i}} = (z(x)-y)\, w_i^f$$

$$\frac{\partial cost}{\partial out_2} = \frac{-y}{z(x)} + \frac{1-y}{1-z(x)} = \frac{-y(1-z(x)) + (1-y)z(x)}{z(x)(1-z(x))} = \frac{z(x)-y}{z(x)(1-z(x))}$$

$$\frac{\partial out_2}{\partial in_2} = z(x)(1-z(x)) \qquad \frac{\partial in_2}{\partial out_{f_i}} = w_i^f$$

$$\frac{\partial out_{f_i}}{\partial in_{f_i}} = \begin{cases} 1 & ; in_{f_i} > 0 \\ 0 & ; in_{f_i} < 0 \end{cases} \qquad \frac{\partial in_{f_i}}{\partial w_{ji}^A} = a(x^{(j)})$$

$$\text{ل) } \frac{\partial cost}{\partial w_{ji}^A} = \begin{cases} (z(x)-y)\, w_i^f\, a(x^{(j)}) & ; in_{f_i} > 0 \\ 0 & ; in_{f_i} < 0 \end{cases}$$

ج) در شکل نشان داده شده که

$$\frac{\partial cost}{\partial out_{f_i}} = (z(x) - y) w_i^F$$

حال داریم

$$\frac{\partial cost}{\partial a_1} = \sum_{i=1}^{20} \frac{\partial cost}{\partial out_{f_i}} \frac{\partial out_{f_i}}{\partial in_{f_i}} \frac{\partial in_{f_i}}{\partial a_1}$$

$$\frac{\partial out_{f_i}}{\partial in_{f_i}} = \begin{cases} 1 & ; in_{f_i} > 0 \\ 0 & ; in_{f_i} < 0 \end{cases} \qquad \frac{\partial in_{f_i}}{\partial a_1} = w_{1i}^A$$

$$\Rightarrow \frac{\partial cost}{\partial a_1} = \begin{cases} \sum_{i=1}^{20} (z(x) - y) w_i^F w_{1i}^A & ; in_{f_i} > 0 \\ 0 & ; in_{f_i} < 0 \end{cases}$$

$$\frac{\partial cost}{\partial out_{c_1}} = \frac{\partial cost}{\partial a_1} \frac{\partial a_1}{\partial out_{c_1}} = \frac{1}{2} \frac{\partial cost}{\partial a_1}$$

$$a_1 = \frac{out_{c_1} + out_{c_2}}{2} \Rightarrow \frac{\partial a_1}{\partial out_{c_1}} = 1/2$$

بنابراین میتوان گفت که :

$$\frac{\partial cost}{\partial out_C} = \frac{1}{2} \begin{bmatrix} \frac{\partial cost}{\partial a_1} \\ \frac{\partial cost}{\partial a_1} \\ \frac{\partial cost}{\partial a_2} \\ \frac{\partial cost}{\partial a_2} \\ \vdots \\ \frac{\partial cost}{\partial a_{car}} \\ \frac{\partial cost}{\partial a_{car}} \end{bmatrix}_{784 \times 1}$$

در نهایت میتوان نوشت :

$$\frac{\partial cost}{\partial w_1^C} = \sum_{i=1}^{784} \frac{\partial cost}{\partial out_{c_i}} \frac{\partial out_{c_i}}{\partial in_{c_i}} \frac{\partial in_{c_i}}{\partial w_1^C}$$

$$\frac{\partial out_{c_i}}{\partial in_{c_i}} = \begin{cases} 1 & ; in_{c_i} > 0 \\ 0 & ; in_{c_i} < 0 \end{cases}$$

0-padding

$$inc_1 = w_1^c x_0 + w_2^c x_1 \Rightarrow \frac{\partial inc_1}{\partial w_1^c} = x_0$$

$$inc_2 = w_1^c x_1 + w_2^c x_2 \Rightarrow \frac{\partial inc_2}{\partial w_1^c} = x_1$$

$$\vdots$$

$$inc_{VAY} = w_1^c x_{VAS} + w_2^c x_{VAY} \Rightarrow \frac{\partial inc_{VAY}}{\partial w_1^c} = x_{VAS}$$

(x_{VAY}: 0-padding)

$$\Rightarrow \frac{\partial inc_i}{\partial w_1^c} = x_{i-1}$$

$$\Rightarrow \frac{\partial cost}{\partial w_1^c} = \begin{cases} \sum_{i=1}^{VAY} \frac{\partial cost}{\partial out_{c_i}} x_{i-1} & ; inc_i > 0 \\ 0 & ; inc_i < 0 \end{cases}$$

با توجه به inc_i داریم:

$$\frac{\partial cost}{\partial w_1^c} = \frac{\partial cost}{\partial out_{c_1}} x_0 + \frac{\partial cost}{\partial out_{c_2}} x_1 + \dots + \frac{\partial cost}{\partial out_{c_{VAY}}} x_{VAS}$$

$$= \frac{1}{2}\left[\frac{\partial cost}{\partial a_1}(x_0 + x_1) + \frac{\partial cost}{\partial a_2}(x_2 + x_3) + \dots + \frac{\partial cost}{\partial a_{295}}(x_{VAE} + x_{VAS})\right]$$

با توجه به inf_i داریم:

$$\frac{\partial cost}{\partial w_1^c} = \frac{1}{2}\left[(x_0 + x_1)\sum_{i=1}^{20} w_i^F w_{1i}^A + (x_2 + x_3)\sum_{i=1}^{20} w_i^F w_{2i}^A + \dots + (x_{VAE} + x_{VAS})\sum_{i=1}^{20} w_i^F w_{295i}^A\right]$$

$(Z(x) - y)$

۶) آ) در هر لایه محدوده دید به میزان $K-1$ از این پیدا می‌کند. از آنجایی که هسته‌ها دارای ابعاد $K \times K$ هستند، محدوده دید داده‌ی شده با $l(K-1)+1$ بعد از l لایه کانولوشن با فیلترهای $K \times K$ خواهد بود.

ب) اگر اندازه فیلتر داده‌ی $F \times F$ باشد و اندازه گام استراید برابر D باشد، مثل این است که اندازه فیلتر از F به $D(F-1)+1$ تبدیل می‌شود. حال اگر اندازه ورودی داده‌ها $M \times N$ باشد، آنگاه اندازه خروجی داده می‌شود با:

$$(M - D(F-1) - 1 + 1) \times (N - D(F-1) - 1 + 1) = (M - DF + D) \times (N - DF + D)$$

ج) به عنوان مثال فرض کنید ماتریس A یک ماتریس $D \times D$ باشد که درایه‌های سطر اول و ستون اول آن داده‌ی یک باشد و سایر درایه‌های آن صفر هستند. به عنوان مثال اگر $D=2$ آنگاه $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. البته دقت کنید حاصل $K \otimes A$ حاصل شده، به $D-1$ سطر و ستون آخر آن که همگی صفر هستند را حذف کنیم.

د) روش‌های Masked convolutions درایه‌های خاص از فیلتر را صفر می‌کنیم (Mask می‌کنند) و به این صورت می‌توان اثر بعضی از پیکسل‌ها را در نظر نگرفت و ارتباط بین سایر پیکسل‌ها را استخراج کرد. در روش چین محدوده دید فیلتر تغییر می‌کند و سعی بعضی از درایه‌های آن صفر می‌شوند، یعنی نمی‌توانند ارتباط بین پیکسل‌های مجاور را به خوبی درک کنند. بنابراین بهتر است از روش dialated convolution استفاده کرد زیرا در این روش اولاً محدوده دید فیلتر را می‌توان بزرگتر کرد و ارتباط بین پیکسل‌هایی که با هم رابطه دارند را بهتر درک کرد.