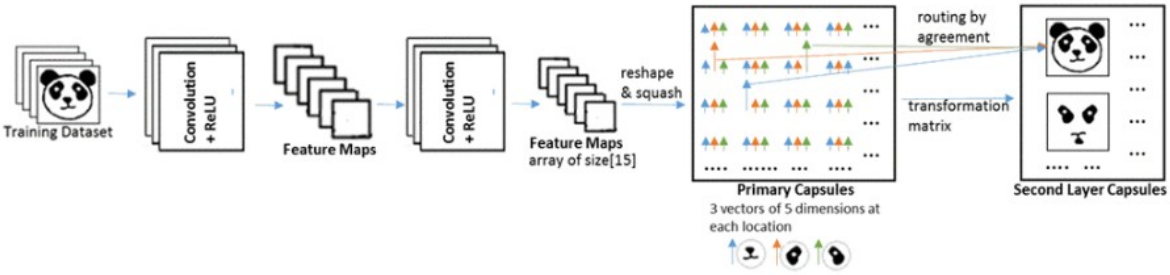
1-الف)

در capsule network ها هر شی را به صورت یک مجموعه از capsule ها نمایش داده می شود. هر capsule مجموعه‌ای از نورون ها است که در کنار همدیگر نمایش دهنده یک ویژگی از آن شی است که این برعکس شبکه های عصبی قبلی است که در آن هر نرون وظیفه تشخیص یک ویژگی به خصوص از شی را داشت. در capsule network ها، هر کپسول مسئول تشخیص یکی از ویژگی های شی است. در کپسول چون به جای یک نورون با تعدادی نورون سر و کار داریم خروجی یک کپسول به جای یک مقدار اسکالر یک برداری از مقادیر است که capsule activation نام دارد و نشان دهنده احتمال حضور یک ویژگی مانند حالت، اندازه، جهت و ... در شی است. در شبکه کپسولی از dynamic routing به جای اتصالات ثابت و static که در شبکه‌های قبلی استفاده می شداستفاده میشود. برای تعیین وزن های فعالسازی در لایه بعدی ما از هماهنگی و توافق فعالسازی های یک کپسول استفاده می کنیم.

برای نشان دادن احتمال وجود یک موجودیت از طول بردار فعالیت کپسول و از جهت گیری آن برای نشان دادن params instantiation استفاده میکنیم. کپسول های فعال در یک سطح، از طریق ماتریس های transformation، پارامتر های instantiation کپسول های سطح بالاتر را پیش بینی میکنند.



حرکت رو به جلو در شبکه:

بر روی داده ورودی چند فیلتر کانولوشن اعمال می‌شود که در نتیجه آن به تعدادی نقشه فعال سازی می رسیم. بعد از آن نقشه های فعال سازی به ۳ بردار تقسیم می‌شوند که هر بردار ۵ بعد دارد. در اینجا هر نقشه نشان دهنده یک ویژگی است. بعد از آن بردار ها بین ۰ تا ۱ برده می‌شوند تا به احتمالات برسیم. سپس به کمک کپسول های لایه فعلی خروجی لایه بعدی پیشبینی می شود.

ب)

شبکه‌های کانولوشن نسب به شیف حساسیت ندارند به گونه‌ای که یک شبکه مناسب می‌تواند هر سه تصویر زیر را به درستی شناسایی کند:

این مهم به دلیل استفاده از پولینگ در شبکه حاصل می شود.

اما به طور ذاتی این شبکه‌ها به دفورمه شدن تصویر حساس نیستند و یا به روتیت شدن آن حساس هستند.

به عبارت دیگر اگر جای اعضای صورت پاندا را در تصاویر بالا با هم جا به جا کنیم باز هم شبکه پاندا را تشخیص میدهد. درواقع شبکه یاد گرفته است که صورت پاندا این دسته از ویژگی‌ها را دارد اما رابطه بین این ویژگی‌ها یادگرفته نشده است. از طرف دیگر اگر شبکه در هنگام یادگیری با چهره چرخیده شده پاندا رو به رو نشود نمی‌تواند در هنگام تست درست آن را پیش‌بینی کند. این محدودیت‌ها در شبکه‌های کپسولی وجود ندارد. این شبکه‌ها میتوانند رابطه بین ویژگی‌ها را نیز یادبگیرند و نسبت به چرخش حساس نباشند. شکل زیر بیان گر این مسیله است:

ج)

شبکه‌های کپسولی برای حل مشکل واریانس دیدگاه در شبکه‌های عصبی کانولوشنال (CNN) به وجود آمدند. کپسول نت یک دیدگاه ثابت است که شامل تغییر ناپذیری چرخشی و انتقالی است.

CNN ها با استفاده از حداکثر ادغام، تغییر ناپذیری ترجمه ای دارند، اما منجر به از دست دادن اطلاعات در زمینه گیرنده می شود. و همانطور که شبکه عمیق تر می شود، میدان دریافت نیز به تدریج افزایش می یابد و از این رو تجمع حداکثری در لایه های عمیق تر باعث از دست رفتن اطلاعات بیشتر می شود. این منجر به از دست رفتن اطلاعات مکانی می شود و تنها اطلاعات محلی/زمانی توسط شبکه یاد می شود.

2-الف)

تابع ادغام، خروجی شبکه در یک موقعیت مشخص را با در نظر گرفتن یک مشخصه آماری مانند جمع یا میانگین با مقادیر در همسایگی آن جایگزین میکند. سایر دلایل استفاده از ادغام: کاهش ابعاد، translation invariance (مدل نسبت به شیفت خوردن تصاویر و دیتا ها حساس نیست) و کاهش زمان محاسبات، Down sampling به جلوگیری از over-fitting کمک می کند.

ادغام حداکثر، ماکسیمم در یک همسایگی را محاسبه می کند (مهم ترین مقدار را نگه میدارد)

ادغام میانگین، میانگین در یک همسایگی را محاسبه می کند (از شرکت تمام داده‌های همسایگی بهره می برد)

ادغام حداکثر خاصیت translation invariance بیشتری نسبت به میانگین دارد زیرا صرفاً از نقاط با اهمیت استفاده می کند. اما ادغام میانگین یه جور اسموسینگ روی نقشه ویژگی اعمال می‌کند که باعث جلو گیری از بیش برازش می شود.

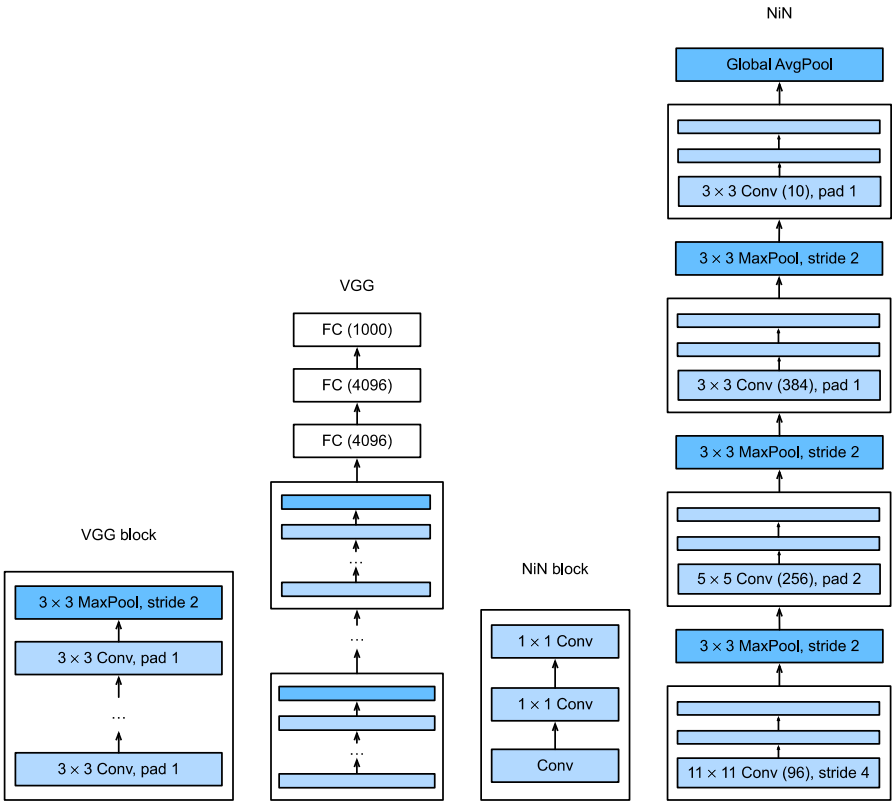
ج)

LeNet، AlexNet، و VGG همگی یک الگوی طراحی مشترک دارند: استخراج ویژگی‌ها با بهره‌برداری از ساختار فضایی از طریق دنباله‌ای از کانولوشن و لایه‌های ادغام شده و پس پردازش نمایش‌ها از طریق لایه‌های کاملاً متصل. بهبودهای LeNet توسط AlexNet و VGG عمدتاً در نحوه گسترش و تعمیق این دو ماژول توسط شبکه‌های بعدی نهفته است. این طراحی دو چالش بزرگ را به همراه دارد. اول، لایه‌های کاملاً متصل در انتهای معماری، تعداد زیادی پارامتر را مصرف می‌کنند. به عنوان مثال، حتی یک مدل ساده مانند VGG-11 به یک ماتریس بسیار بزرگ نیاز دارد که تقریباً 400 مگابایت رم را اشغال می کند. دوم، اضافه کردن لایه‌های کاملاً متصل زودتر به شبکه برای افزایش درجه غیرخطی به همان اندازه غیرممکن است: انجام این کار ساختار فضایی را از بین می‌برد و به طور بالقوه حتی به حافظه بیشتری نیاز دارد.

شبکه در بلوک های شبکه (NiN) جایگزینی را ارائه می دهد که قادر به حل هر دو مشکل در یک استراتژی ساده است. آنها بر اساس یک راهکار بسیار ساده پیشنهاد شدند: (1) استفاده از کانولوشن برای اضافه کردن غیرخطی‌های محلی در سراسر فعال‌سازی کانال و (ب) استفاده از ادغام میانگین جهانی برای ادغام در تمام مکان‌ها در آخرین لایه نمایش.

ورودی و خروجی لایه‌های کانولوشن شامل تنسور های چهار بعدی با محورهای مربوط به مثال، کانال، ارتفاع و عرض است. همچنین به یاد داشته باشید که ورودی و خروجی لایه‌های کاملاً متصل معمولاً تنسور های دو بعدی هستند که مطابق با مثال و ویژگی هستند. ایده پشت NiN این است که یک لایه کاملا متصل در هر مکان پیکسل (برای هر ارتفاع و عرض) اعمال شود. پیچیدگی حاصل را می توان به عنوان یک لایه کاملاً متصل در نظر گرفت که به طور مستقل روی هر مکان پیکسل عمل می کند.

شکل زیر تفاوت های ساختاری اصلی بین VGG و NiN و بلوک های آنها را نشان می دهد.



تعداد پارامتر ها:

GoogleNet: 11,193,984

VGG-16: 138,000,000

AlexNet: 62,300,000

NiN: 1,769,112

NiN پارامترهای بسیار کمتری نسبت به AlexNet و VGG دارد. این در درجه اول از این واقعیت ناشی می شود که نیازی به لایه های غول پیکر کاملاً متصل ندارد. در عوض، از ادغام میانگین جهانی برای جمع‌آوری در تمام مکان‌های تصویر پس از آخرین مرحله بدنه شبکه استفاده می‌کند. این امر نیاز به عملیات کاهش گران قیمت (یادگیری شده) را از بین می برد و آنها را با میانگین ساده جایگزین می کند..

اگر شبکه ای با لایه های عمیق زیاد ساخته شود، ممکن است با مشکل بیش از حد برازش مواجه شود. برای حل این مشکل، نویسندگان در مقاله تحقیقاتی Going deeper with convolutions، معماری GoogleNet را با ایده داشتن فیلترهایی با اندازه های متعدد که می توانند در یک سطح کار کنند، پیشنهاد کردند. با این ایده، شبکه در واقع گسترده تر می شود تا عمیق تر. از آنجایی که آموزش شبکه‌های عصبی زمان‌بر و پرهزینه است، نویسندگان مقاله تعداد کانال‌های ورودی را با افزودن یک کانولوشن اضافی (1×1) قبل از فیلتر (3×3) و (5× 5) آن‌ها را محدود می‌کنند تا ابعاد کانال را کاهش دهند. که این مهم باعث کاهش قابل توجه پارامتر های این شبکه نسبت به الکس نت و وی جی جی است.

1	9	6	4
5	4	7	8
5	1	2	9
6	7	6	0

4 x 4

max pooling

s = 2

9	8
7	9

2 x 2

9	1	7	4
5	6	3	0
1	2	5	4
0	8	9	0

max pooling

s = 2

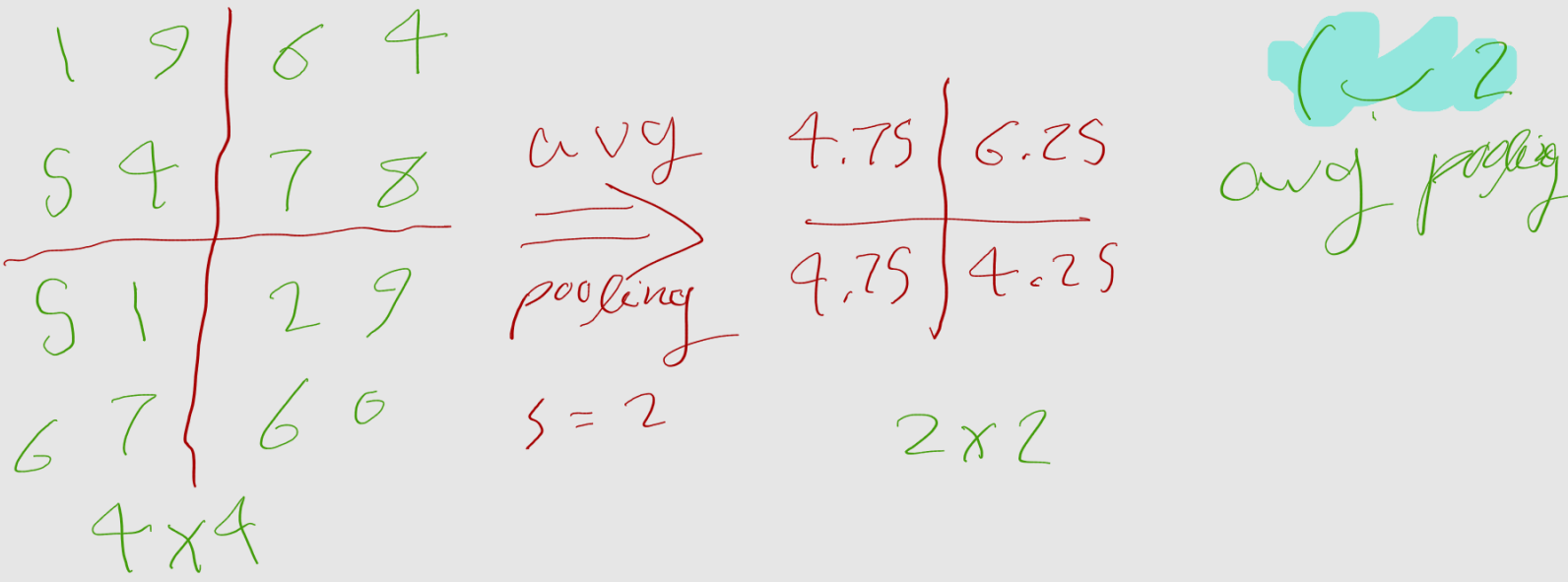
9	7
8	9

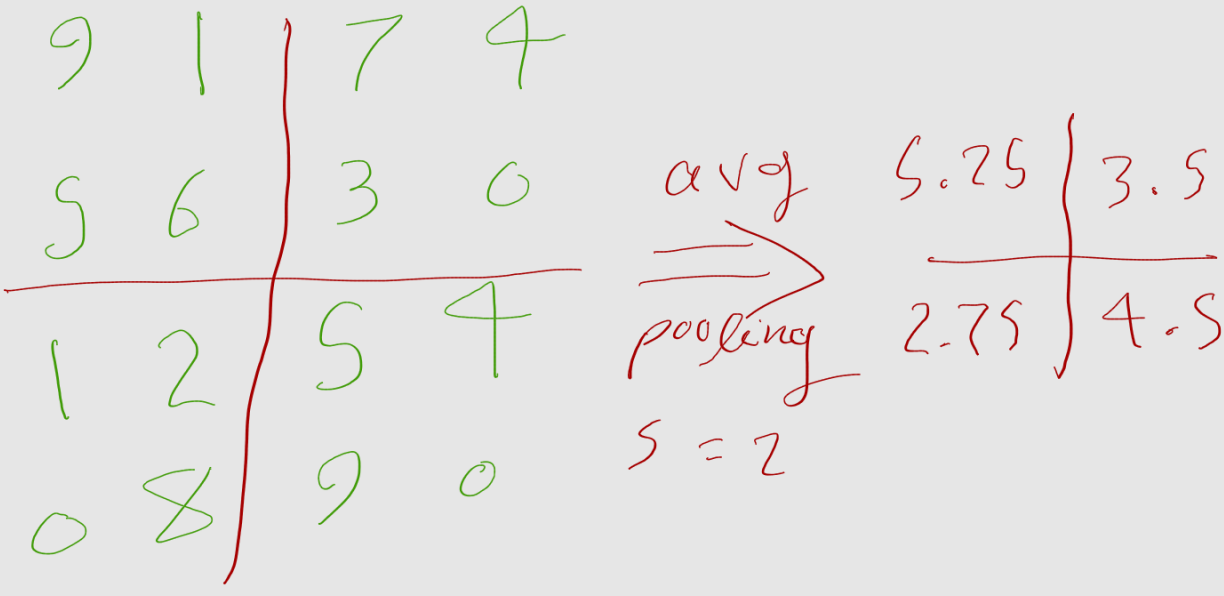
7	6	9	1
5	2	0	4
5	8	3	9
0	2	2	1

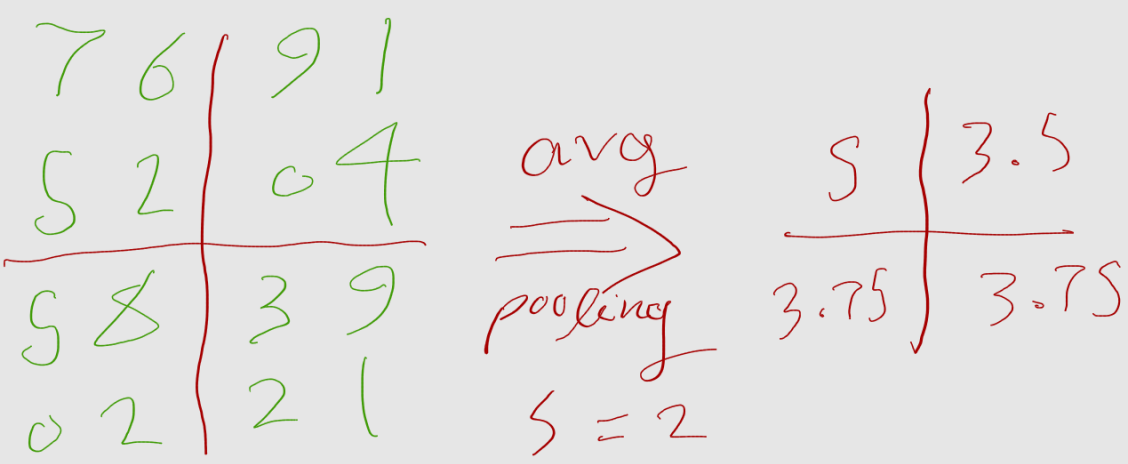
max pooling

s = 2

7	9
8	9


1 9 6 4
5 4 7 8
5 1 2 9
6 7 6 0
4x4
avg
pooling
s = 2
4.75 6.25
4.75 4.25
2x2
avg pooling


9 1 7 4
5 6 3 0
1 2 5 4
0 8 9 0
avg
pooling
s = 2
5.25 3.5
2.75 4.5


7 6 9 1
5 2 0 4
5 8 3 9
0 2 2 1
avg
pooling
s = 2
5 3.5
3.75 3.75

activation map dim formula

$$h_{out} = \left\lfloor \frac{h_{in} + 2p - d \times (k-1) - 1}{s} + 1 \right\rfloor$$

↳ floor

padding — same $= \lfloor \frac{k}{2} \rfloor$

valid $= 0$

$l_1 \longrightarrow h_{in} = 256, p = 1, d = 1, k = 3, s = 1$

$$h_{out} = \left\lfloor \frac{256 + 2 - 2 - 1}{1} + 1 \right\rfloor = 256$$

$l_2 \longrightarrow h_{in} = 256, p = 0, d = 2, k = 5, s = 2$

$$h_{out} = \left\lfloor \frac{256 + 0 - 8 - 1}{2} + 1 \right\rfloor = 124$$

$l_3 \longrightarrow$ max pool (2x2) $s = 2$

↳ halves the input

$h_{in} = 124 \longrightarrow h_{out} = 62$

$l_4 \longrightarrow h_{in} = 62, p = 1, d = 1, k = 3, s = 1$

$$h_{out} = \left\lfloor \frac{62 + 2 - 2 - 1}{1} + 1 \right\rfloor = 62$$

$l_5 \rightarrow h_{in} = 62$, $p=0$, $d=4$, $k=5$, $s=2$

$$h_{out} = \left\lfloor \frac{62 + 0 - 16 - 1}{2} + 1 \right\rfloor = 23$$

$l_6 \rightarrow$ max pool (2×2) $s=2$

$h_{in} = 23 \rightarrow h_{out} = 11$

$l_7 \rightarrow h_{in} = 12$, $p=1$, $d=1$, $k=3$, $s=1$

$$h_{out} = \left\lfloor \frac{11 + 2 - 2 - 1}{1} + 1 \right\rfloor = 11$$

$l_8 \rightarrow h_{in} = 12$, $p=0$, $d=2$ $k=5$, $s=2$

$$h_{out} = \left\lfloor \frac{12 + 0 - 8 - 1}{1} + 1 \right\rfloor = 2$$

$l_9 \rightarrow$ max pool (2×2) $s=2$

$h_{in} = 2 \rightarrow h_{out} = 1$

channels →

$3 \xrightarrow{l_1} 64 \xrightarrow{l_2} 32 \xrightarrow{l_3} 32$

$32 \xrightarrow{l_4} 128 \xrightarrow{l_5} 64 \xrightarrow{l_6} 64$

$64 \xrightarrow{l_7} 256 \xrightarrow{l_8} 128 \xrightarrow{l_9} 128$

x_{11}	x_{12}	x_{13}
x_{21}	x_{22}	x_{23}
x_{31}	x_{32}	x_{33}

X

c_{11}	c_{12}
c_{21}	c_{22}

C

$$X * C = \begin{array}{c|c} o_{11} & o_{12} \\ \hline o_{21} & o_{22} \end{array}$$

$$o_{11} = c_{11}x_{11} + c_{12}x_{12} + c_{21}x_{21} + c_{22}x_{22} + b$$

$$o_{12} = c_{11}x_{12} + c_{12}x_{13} + c_{21}x_{22} + c_{22}x_{23} + b$$

$$o_{21} = c_{11}x_{21} + c_{12}x_{22} + c_{21}x_{31} + c_{22}x_{32} + b$$

$$o_{22} = c_{11}x_{22} + c_{12}x_{23} + c_{21}x_{32} + c_{22}x_{33} + b$$

$$O = \frac{1}{4}(o_{11} + o_{12} + o_{21} + o_{22})$$

$$\frac{\partial l}{\partial c_{11}} = \left[\frac{\partial l}{\partial o} \times \frac{\partial o}{\partial o_{11}} \times \frac{\partial o_{11}}{\partial c_{11}}\right] +$$

$$\left[\frac{\partial l}{\partial o} \times \frac{\partial o}{\partial o_{12}} \times \frac{\partial o_{12}}{\partial c_{11}}\right] +$$

$$\left[\frac{\partial l}{\partial o} \times \frac{\partial o}{\partial o_{21}} \times \frac{\partial o_{21}}{\partial c_{11}}\right] +$$

$$\left[\frac{\partial l}{\partial o} \times \frac{\partial o}{\partial o_{22}} \times \frac{\partial o_{22}}{\partial c_{11}}\right]$$

$$= 1 \times \frac{1}{4} \times \left[x_{11} + x_{12} + x_{21} + x_{22}\right]$$

$$= \frac{1}{4}(1 + 2 - 1 + 5) = \frac{7}{4}$$

$$\frac{\partial l}{\partial c_{12}} = \frac{1}{4}(x_{12} + x_{13} + x_{22} + x_{23}) = \frac{1}{4}(2 - 2 + 5 + 3) = 2$$

$$\frac{\partial l}{\partial c_{21}} = \frac{1}{4}(-1 + 5 + 3 + 0) = \frac{7}{4}$$

$$\frac{\partial l}{\partial C_{22}} = \frac{1}{4}(5+3+0+1) = \frac{9}{4}$$

$$\text{loss} = \frac{7}{4} + 2 + \frac{7}{4} + \frac{9}{4} = \frac{7+8+7+9}{4}$$

$$\hookrightarrow = \frac{31}{4}$$