به نام خداوند بخشنده و مهربان

یادگیری عمیق

تمرین پنجم

محسن نقی‌پورفر ۹۴۱۰۶۷۵۷

۱ Regularization

۱.۱ BatchNormalization

۱.۱.۱ تاثیر اضافه کردن BatchNormalization

در صورت اضافه کردن لایه BatchNormalization تاثیرات زیر به شبکه اعمال می‌شود:

- سرعت آموزش شبکه با توجه به حل شدن مسئله Internal Covariate Shift زیاد می‌شود.
- مسئله وابستگی ورودی‌های هرلایه و لایه بعدی حل می‌شود و این باعث ایجاد امکان یادگیری و بهینه‌سازی شبکه با استفاده از learning rate های بالا می‌شود.
- با حل مسئله ICS،می‌توان از شبکه‌های عمیق‌تر نیز برای مسئله خاص استفاده کرد و به نتایج معمولا بهتری نسبت به شبکه با عمق کمتر دست یافت.
- این لایه، باعث نرمالیزه کردن توزیع خروجی هر لایه در هر نورون می‌شود و این نیز باعث سرعت بخشیدن به فرآیند یادگیری می‌شود.
- با استفاده از آماره‌های Batch، باعث می‌شود که هر شبکه منظم شود و Regularize شود و این منظم شدن را نسبت به Overfitting می‌گوییم. پس این لایه باعث منظم سازی شبکه و در نتیجه جلوگیری از Overfitting می‌شود.
- افزودن این لایه به شبکه اجازه مقداردهی اولیه ضعیف(Poor Initialization) به وزن‌ها و پارامترهای یادگیری شبکه را می‌دهد بدون آنکه در فرآیند یادگیری مشکلی ایجاد شود.
- مسئله explodeشدن و یا vanishشدن گرادیان را به حد قابل قبولی حل می‌کند.
- این لایه با توجه به محدود کردن ورودی تابع فعال‌ساز در ناحیه خاصی که برای تابع فعال ساز relevantمی‌باشد، تا حدی از اشباع شدن خروجی تابع جلوگیری می‌کند و به نوعی شبکه را stablize می‌کند.
- اختلافات خطی بین Batchهای مختلف را از بین می‌برد.
- ارزش و توانایی مدل را با اضافه کردن دو پارامتر قابل یادگیری γ و β افزایش می‌دهد.
- در زمان تست با توجه به اینکه ممکن است Batch برای داده‌های تست خیلی کوچک باشد، با استفاده EMA تخمین خوبی از میانگین و واریانس Batch می‌زند.

۲.۱.۱ تعداد پارامتر های افزوده شده

در اثر اضافه کردن لایه BatchNormalization به هر لایه، به ازای هر نورون آن لایه، دو پارامتر γو β اضافه می‌شود که قابل یادگیری هستند.

۳.۱.۱ پیاده سازی تابع این لایه

عکس‌های زیر، پیاده سازی لایه BatchNormalization می‌باشند. همانطور که از اسم آنها نیز پیداست، عکس اول مربوط به پیاده‌سازی این لایه برای بعد از لایه Convolution است و عکس دوم مربوط به پیاده‌سازی این لایه بعد از یک لایه Fully Connected می‌باشد.

۲.۱ Dropout

۱.۲.۱ تاثیر اضافه کردن Dropout

اضافه کردن این منظم‌ساز به هر لایه، تاثیرات زیر را همراه خود می‌آورد:

- تاثیر مشخص اول، همان منظم‌ساز بودن آن است، به این معنی که از overfitting جلوگیری می‌کند.
- باتوجه به منظم‌ساز بودن آن، به feature co-adaptation کمک می‌شود.
- باتوجه به اینکه منظم ساز می‌باشد، به فرآیند یادگیری، نویز اضافه می‌کند که این یکی از تاثیرات اضافه کردن منظم‌ساز است.
- باعث بیشتر شدن Sparsity در لایه‌های پنهان شبکه می‌شود که این نیز از تاثیرات اضافه کردن این منظم‌ساز است.
- در واقع شبیه میانگین گیری از تمام 2^H حالت ممکن برای مدل‌ها می‌باشد(در صورتی که به یک لایه پنهان با H نورون اعمال شود.). در این حالت وزن‌ها به نوعی Shared هستند و این باعث منظم شدن مدل شده و از overfitting جلوگیری می‌کند.

۲.۲.۱ فرق در آموزش و تست

در هنگام آموزش، ما از یک توزیع Binomial با احتمال موفقیت Pاستفاده می‌کنیم و نورون‌های لایه‌ای که این منظم‌ساز روی آن اعمال شده است را به طور کامل حذف می‌کنیم. در واقع هر نورون با احتمال P در شبکه حضور دارد و یک سری از نورون‌ها در فرآیند یادگیری حذف می‌شوند. اما در فرآیند تست، همه نورون‌ها در شبکه به طور قطع حاضر هستند، اما وزن Connectionآنها در عدد P ضرب می‌شود و به نوعی Expected گیری انجام می‌دهیم برای خروجی هر نورون و تاثیرات آن روی نورون‌های لایه بعدی. یعنی، انتظار داریم که خروجی نورون که با احتمال P از شبکه در فرآیند یادگیری حذف می‌شد،خروجی آن به ازای وزن‌های Pw باشد. شکل زیر که از مقاله خود جناب آقای Hinton از مقاله Dropout آمده است، فرق بین اعمال این منظم ساز در فرآیند تست و آموزش را به خوبی توضیح می‌دهد.

۳.۲.۱ پیاده سازی تابع این لایه

در عکس زیر پیاده‌سازی مربوط به این منظم‌ساز روی ورودی خود آمده است.

۲ Google Colab

۱.۲ گزارش نتیجه و مراحل اجرا در این محیط

۲.۲ مقایسه در حالت وجود یا عدم وجود منظم سازها

۳.۲ گزارش نتیجه در اثر وجود دو منظم ساز

۳ Visualization

۱.۳ توضیح در مورد شبکه VGG

این شبکه در سال ۲۰۱۴ در کنفرانس ICLR معرفی شد. این شبکه دو نوع VGG16 و VGG19 دارد که به ترتیب از ۱۶ و ۱۹ لایه تشکیل شده‌اند. در این شبکه‌ها فیلتر های وزن در لایه‌های کانوولوشنی بسیار کوچک می‌باشند و در سایز ۳ در ۳ می‌باشند که طبق گفته مقاله این شبکه، این فیلترها باعث عمیق‌تر کردن شبکه و عین حال نتیجه بهتر نسبت به مدل‌های مشابه گرفتن، می‌باشند. در واقع وجود این فیلترها باعث شده تا تعداد لایه های شبکه تا ۱۶ یا ۱۹ لایه پیش‌برود و دقت آن نیز نسبت به مدل‌های مشابه بهتر باشد. این شبکه برنده مسابقه IMAGENET Challenge در تسک Localization در سال ۲۰۱۴ و برنده مقام دومی در تسک Classification در همان سال می‌باشد. این مسابقه که هرساله برگزار می‌شود دارای دیتای بسیار معروفی به نام ILSVRC باشد. این شبکه با بیشتر کردن عمق خود با استفاده از فیکس کردن سایز فیلترهای وزن لایه‌های کانوولوشن و بسیار کوچک بودن سایز آن سعی در بهتر کردن دقت خود داشته است و در این زمینه نیز موفق بوده است. طبق گفته نویسندگان این مقاله، این شبکه نه تنها برای دیتای مسابقه ILSVRC بسیار خوب عمل می‌کند بلکه روی دیتای مسابقه‌های دیگر نیز بسیار خود عمل کرده است. این شبکه علاوه بر دیتای ILSVRC در مقاله خود بر روی دیتاهای VOC-2012،VOC-2007، Caltech-101 و Caltech-256 نیز برای تسک های Classification و Localization تست شده است. نتایج مربوط به دقت این شبکه در جداول زیر آمده است. این شبکه برای ارزیابی روی دیتای ILSVRC از دو معیار Top-1 Error و Top-5 Error استفاده کرده است که اولی نسبت تعداد عکس های به اشتباه طبقه‌بندی شده در داده تست است ولی دومی نسبت تعداد عکس‌هایی به کل است که کلاس درست برای این عکس‌ها در ۵ کلاس اول محتمل که شبکه پیش‌بینی کرده است می‌باشد. بنابراین مشخص است که خطای Top-5 نسبت به خطای Top-1 همیشه مقدار کمتری برای شبکه‌های مختلف خواهد داشت. علت وجود این نوع جدید از خطا نیز وجود ۱۰۰۰ کلاس در دیتای ILSVRC می‌باشد که تعداد خیلی زیادی است و شهود آن به این معنی است که اگر شبکه برای عکس ورودی از بین ۱۰۰۰ کلاس، کلاس درست را در ۵ کلاس محتمل‌ترین برای یک عکس ببیند، قابل قبول است و خطا نیست. نتایج مربوط به این خطا نیز برای این دیتا در جدول زیر آمده است.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

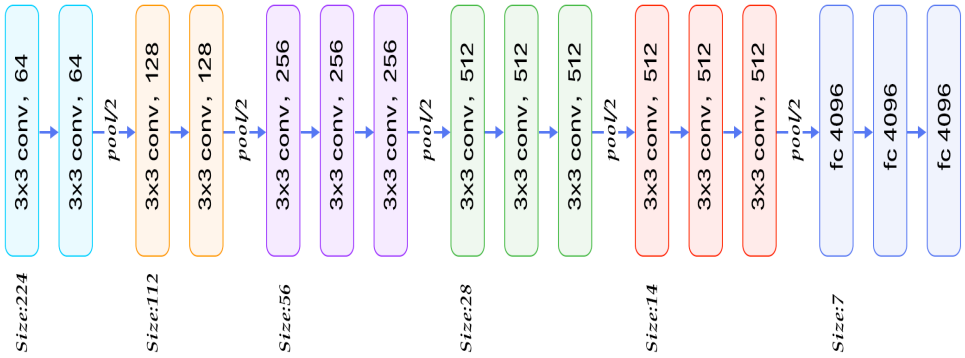
شکل ۱: جدول مربوط نتایج شبکه VGG روی دیتای ILSVRC

Method	VOC-2007 (mean AP)	VOC-2012 (mean AP)	Caltech-101 (mean class recall)	Caltech-256 (mean class recall)
Zeiler & Fergus (Zeiler & Fergus, 2013)	-	79.0	86.5 ± 0.5	74.2 ± 0.3
Chatfield et al. (Chatfield et al., 2014)	82.4	83.2	88.4 ± 0.6	77.6 ± 0.1
He et al. (He et al., 2014)	82.4	-	$\mathbf{93.4 \pm 0.5}$	-
Wei et al. (Wei et al., 2014)	81.5 (85.2*)	81.7 (**90.3***)	-	-
VGG Net-D (16 layers)	89.3	89.0	91.8 ± 1.0	85.0 ± 0.2
VGG Net-E (19 layers)	89.3	89.0	92.3 ± 0.5	85.1 ± 0.3
VGG Net-D & Net-E	**89.7**	**89.3**	92.7 ± 0.5	$\mathbf{86.2 \pm 0.3}$

شکل ۲: جدول مربوط نتایج شبکه VGG روی دیتاهای دیگر

۲.۳ توضیح معماری شبکه VGG

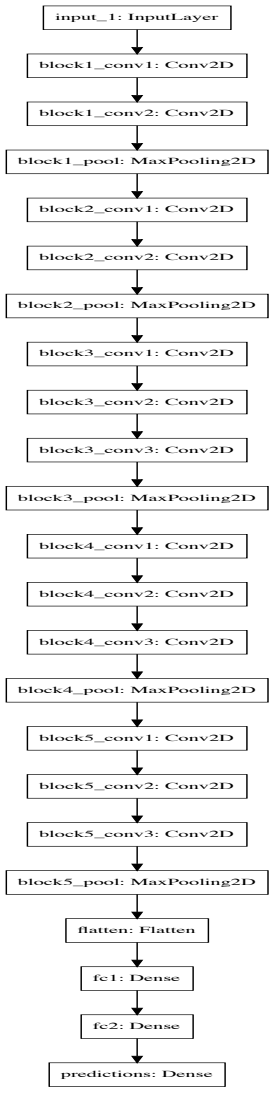
این شبکه از ۱۶ لایه تشکیل شده است. عکس مربوط به معماری شبکه و پارامترهای هر لایه و ابعاد هر لایه در عکس دوم قابل مشاهده است. این شبکه در کل حدود ۱۳۸ میلیون پارامتر قابل یادگیری دارد. همچنین نوع لایه‌ها در عکس دوم و اول قابل مشاهده است.



شکل ۳: معماری شبکه VGG16

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0



شکل ۴: ابعاد وزن‌ها و ورودی و تعداد پارامترهای قابل یادگیری در لایه‌های مختلف در شبکه VGG16

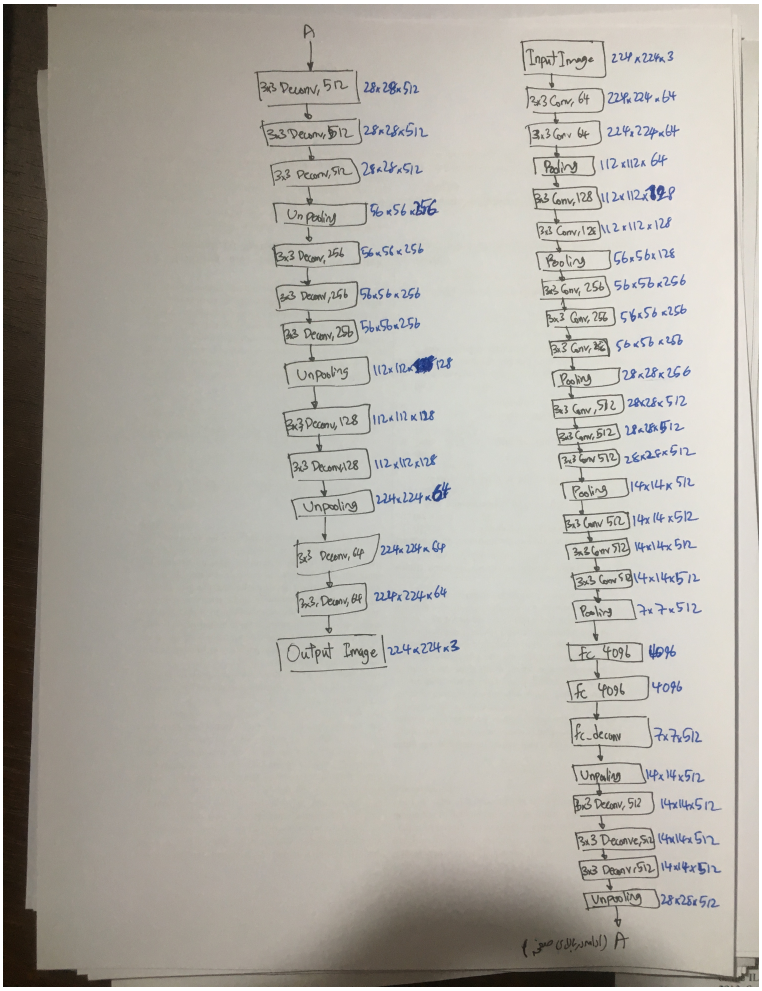
۳.۳ گزارش فیلترهای هر لایه و مقایسه اولین و آخرین فیلتر

۴.۳ تحلیل نتایج لایه‌های ۳ و ۱۳ حاصل از ورودی های جدید شبکه

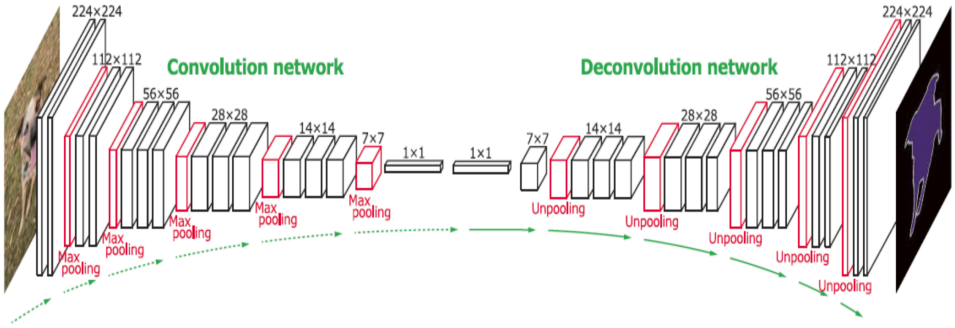
۴ DeConvolution

۱.۴ رسم شبکه عصبی و مشخصات هر لایه

همانطور که از کد مشخص است، این شبکه، در قسمت encoder همان شبکه VGG16 می‌باشد که معماری و مشخصات آن در سوال دوم کشیده و توضیح داده شد. این شبکه در قسمت Decoder خود، دقیقا برعکس VGG عمل می‌کند یعنی به ازای هر لایه Convolution یک لایه Deconvolution و به ازای هر لایه Pooling یک لایه Unpooling گذاشته شده است و در نهایت در خروجی شبکه، سایز تصویر برابر سایز ورودی شبکه می‌باشد.



شکل ۵: عکس مربوط به رسم شبکه‌عصبی موجود در کد net.py که همان شبکه معروف Deconvnet است.



شکل ۶: عکس مربوط به شبکه Deconvnet در اینترنت

۲.۴ کاربر در شبکه های عمیق

۳.۴ نحوه عملکر این لایه و تفاوت با لایه Convolution