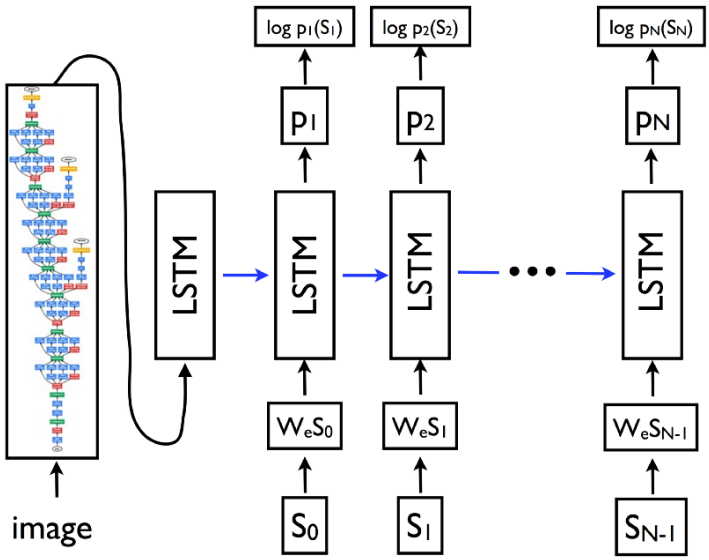
عنوان: Image Captioning

در این پروژه، شما قرار است شبکه‌ای طراحی کنید که به عنوان ورودی، یک تصویر را می‌گیرد و در خروجی خود، یک **caption** مناسب (یعنی توضیحی که اولا درست باشد و دوما بیشترین جزییات ممکن از تصویر را در بر بگیرد) تولید می‌کند. در طول درس، با کلیت **image captioning** آشنا شده‌اید. از کاربردهای این مساله می‌توان به موارد زیر اشاره کرد:

۱. توضیح تصویر به صورت اتوماتیک برای افراد نابینا (با کمک تبدیل متن تولید شده به گفتار)
۲. امکان مدیریت **content** بهتر عکس ها در فضای مجازی و مرورگرها، برای مثال پیشنهاد تصاویر مرتبط در **image search** گوگل
۳. ماشین های خودران، برای توصیف بهتر محیط به عنوان ورودی قسمت تصمیم‌گیرنده آن

یک ساختار تولید **caption** برای تصویر، از دو شبکه تشکیل می‌شود. یک شبکه‌ی **CNN** و یک شبکه‌ی **RNN** (که از مزیت‌های این ساختار این است که از قدرت هر دو شبکه استفاده می‌کند). شبکه‌ی **CNN**، اطلاعات فضایی و ویژگی های تصویر را استخراج می‌کند (این بخش را با نام‌های **feature extractor** و **Encoder** و **Embedding** نیز می‌شناسند) و شبکه‌ی **RNN**، دنباله‌ی واژگان را برای ما می‌سازد (این بخش را نیز با نام **Decoder** می‌شناسند). برای مثال، قسمت **CNN** می تواند یک **VGG16** و قسمت **RNN** می‌تواند یک **LSTM** باشد. کلیت شماتیک یک نمونه از این شبکه‌ها به شکل زیر است:



در این پروژه، مجموعه داده‌ای که در اختیار شما قرار داده شده است شامل تصاویر، برچسب اشیا موجود در تصویر به همراه جملات توصیف کننده تصویر مورد نظر خواهد بود. تصاویر داده شده شامل دو زیر بخش یادگیری و ارزیابی خواهد بود، همچنین بخشی از تصاویر نیز برای تست نتایج در روز تحویل پروژه در نظر گرفته شده‌اند.

پردازش‌های ابتدایی:

برای استفاده از کپشن‌ها در فرآیند آموزش شبکه، لازم است پیش پردازش‌هایی روی آنها انجام دهید. تعدادی از آنها موارد زیرند:

۱. تبدیل کپشن‌ها به حروف کوچک (**lower case**)

۲. حذف علائم نگارشی (، و . و : و ...)

۳. حذف اعداد و کاراکترهای نامناسب (**#, %, &, Ö**) از کپشن‌ها

۴. حذف فاصله و نیم‌فاصله

۵. (دلخواه) حذف کاراکترهای تک حرفی (مانند **a**) از کپشن‌ها

شبکه‌ی استخراج ویژگی:

همانطور که توضیح داده شد، شبکه از دو قسمت تشکیل می‌شود که قسمت اول آن وظیفه‌ی استخراج ویژگی تصاویر را برعهده دارد. پس نیاز داریم تا یک شبکه‌ی کانولوشنی طراحی کنیم تا این عمل را برای ما انجام دهد. برای این کار، ابتدا شبکه‌ای برای برچسب زدن روی تصاویر طراحی می‌کنیم و سپس، لایه‌ی انتهایی آن (لایه‌ی تمام متصل و **softmax** برای برچسب زدن به کلاس) را از آن جدا می‌کنیم و از لایه‌ی **feature vector** آن استفاده می‌کنیم.

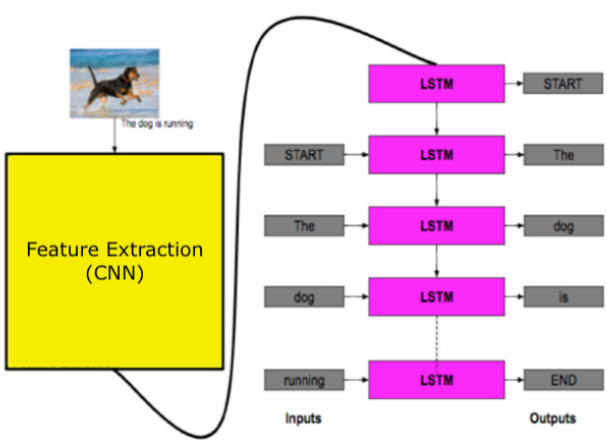
بخش اول:

یک شبکه طراحی کنید و از آن برای برچسب زدن به تصاویر استفاده کنید. در انتخاب نوع و پارامترهای شبکه کاملا مختار هستید.(۳۵)

ارزیابی بخش اول: صحت طبقه‌بندی روی مجموعه تصاویری که هنگام ارائه آن در اختیار شما قرار میگیرد ، روی آن اجرا می‌کنید، معیار نمره‌دهی این بخش است (دقت شبکه‌ی شما روی داده‌های تستی که در اختیار شما قرار گرفته‌است، می‌تواند حدود مناسبی از این دقت را نشان دهد. به هیچ عنوان از داده‌های تست برای آموزش شبکه استفاده نکنید). دقت شبکه شما برای نمره‌ی کامل باید حداقل ۹۰ درصد باشد. دقت‌های بالای ۹۰ درصد، مشمول نمره‌ی امتیازی می‌شوند. (امکان تغییر این عدد در صورت پیچیدگی بالای مساله و عدم دستیابی تعداد زیادی از دانشجویان به این دقت، خواهد بود) (۱۰)

در ادامه، به سراغ کامل کردن شبکه و **caption** زدن می‌رویم. به کمک شبکه‌ی طراحی شده، می‌توانیم بردار ویژگی‌های هر تصویر را استخراج کنیم (که برای مثال برداری ۲۰۴۸ تایی است). می‌توانید نیمه‌ی **CNN** شبکه را نیز بعد از اضافه کردن نیمه‌ی **RNN** مجددا آموزش داده و بهبود دهید ولی اگر می‌خواید آموزش شما در قسمت **RNN** سریع‌تر باشد و بتوانید تغییرات سریع‌تری را روی آن اعمال کنید، می‌توانید نیمه‌ی **CNN** را تغییر ندهید و به عنوان پیش پردازش بخش بعد، همه‌ی تصاویر را به **feature vector** تبدیل کنید و در جایی به عنوان **input** برای قسمت‌های بعد ذخیره کنید (این کار دقت نهایی شما را ممکن است کاهش دهد ولی با سرعت بیشتری می‌توانید قسمت **RNN** را روی این حجم داده آموزش دهید).

شبکه‌ی بازگشتی قرار است واژه به واژه، **caption** مطلوب برای تصاویر را تولید کند:



که عمل **RNN**، چیزی مشابه جدول زیر است:

	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	startseq	the
2	Image_1	startseq the	black
3	Image_1	startseq the black	cat
4	Image_1	startseq the black cat	sat
5	Image_1	startseq the black cat sat	on
6	Image_1	startseq the black cat sat on	grass
7	Image_1	startseq the black cat sat on grass	endseq

(کاراکتر ورودی و خروجی دلخواهند. برای آن که با واژه‌های معمول قاطی نشوند، از **START** و **END** یا **startseq** یا **endseq** می‌توان استفاده کرد).

اما از آنجا که شبکه‌ی ما، واژه به واژه و نه حرف به حرف **caption** را تولید می‌کند، برای تبدیل واژگان به عدد (برای قابل فهم شدن برای شبکه) نیاز به استفاده از یک دیکشنری و یا **wors2vec** داریم. این دیکشنری هر کلمه را به یک عدد تبدیل می‌کند. پس جدول **RNN** ما چیزی مانند زیر می‌شود:

	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	[9]	10
2	Image_1	[9, 10]	1
3	Image_1	[9, 10, 1]	2
4	Image_1	[9, 10, 1, 2]	8
5	Image_1	[9, 10, 1, 2, 8]	6
6	Image_1	[9, 10, 1, 2, 8, 6]	4
7	Image_1	[9, 10, 1, 2, 8, 6, 4]	3
8	Image_2	[9]	10
9	Image_2	[9, 10]	12
10	Image_2	[9, 10, 12]	2
11	Image_2	[9, 10, 12, 2]	5
12	Image_2	[9, 10, 12, 2, 5]	11
13	Image_2	[9, 10, 12, 2, 5, 11]	6
14	Image_2	[9, 10, 12, 2, 5, 11, 6]	7
15	Image_2	[9, 10, 12, 2, 5, 11, 6, 7]	3

(در این مثال، **startseq=9** و **endseq=3** است). در این پروژه، شما نیاز به طراحی دیکشنری ندارید و می‌توانید از یک **word2vec** آماده استفاده کنید (می‌توانید خودتان به واژگان موجود در کپشن‌ها عدد نسبت بدهید اما در این صورت، نزدیکی مفهومی واژگان با عدد نزدیک به هم را از دست می‌دهید). برای آنکه امکان آموزش شبکه به صورت **Batch** را نیز داشته باشید باید اندازه‌ی ورودی‌ها را یکسان کنید. یعنی به صورت شماتیک، چیزی مانند شکل زیر (که **zero-pad** شده است):

	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	[9, 0, 0 …., 0]	10
2	Image_1	[9, 10, 0, 0 …., 0]	1
3	Image_1	[9, 10, 1, 0, 0 …., 0]	2
4	Image_1	[9, 10, 1, 2, 0, 0 …., 0]	8
5	Image_1	[9, 10, 1, 2, 8, 0, 0 …., 0]	6
6	Image_1	[9, 10, 1, 2, 8, 6, 0, 0 …., 0]	4
7	Image_1	[9, 10, 1, 2, 8, 6, 4, 0, 0 …., 0]	3
8	Image_2	[9, 0, 0 …., 0]	10
9	Image_2	[9, 10, 0, 0 …., 0]	12
10	Image_2	[9, 10, 12, 0, 0 …., 0]	2
11	Image_2	[9, 10, 12, 2, 0, 0 …., 0]	5
12	Image_2	[9, 10, 12, 2, 5, 0, 0 …., 0]	11
13	Image_2	[9, 10, 12, 2, 5, 11, 0, 0 …., 0]	6
14	Image_2	[9, 10, 12, 2, 5, 11, 6, 0, 0 …., 0]	7
15	Image_2	[9, 10, 12, 2, 5, 11, 6, 7, 0, 0 …., 0]	3

بخش دوم:

یک شبکه بازگشتی (با ساختار و پارامترهای کاملا دلخواه) طراحی کنید و از آن برای کپشن زدن به تصاویر استفاده کنید. آن را با تابع هزینه‌ی دلخواه آموزش دهید. (۴۵)

ارزیابی بخش دوم: ارزیابی این بخش بر اساس معنادار بودن جملات تولید شده توسط شبکه بر روی تصاویر تست (که در روز ارائه در اختیارتان قرار می‌گیرد) خواهد بود. (۱۰)

نمره امتیازی: پیاده سازی شبکه خود را همراه با **Attention** انجام دهید (در صورت پیاده سازی با **Attention**، نیازی به پیاده سازی معمولی شبکه نیست). (۱۰). سایر ایده‌ها و روش‌های خلاقانه و بهبود دهنده‌ی در طراحی شبکه‌ها نیز نمره‌ی امتیازی خواهد داشت.

نمره امتیازی: تحویل پروژه حداقل یک هفته پیش از زمان تحویل پایانی شامل نمره امتیازی خواهد شد.