



دانشکده مهندسی کامپیوتر

دکتر محمدرضا محمدی

بهار ۱۴۰۰

تمرین سری سیزدهم

یادگیری عمیق

مجتبی نافذ 96431335

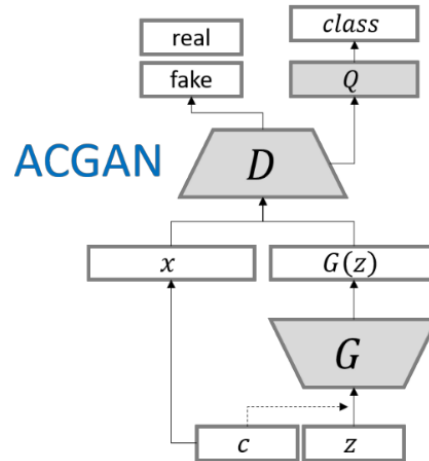
مهلت تحویل : ۲۴ خرداد ۱۴۰۰ ساعت ۲۳:۵۹:۵۹



حال فرض کنید در یک episode به ترتیب عمل‌های زیر انجام شده است: بالا، راست، راست، پائین.

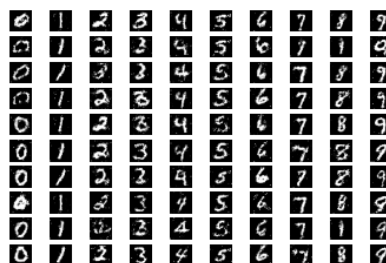
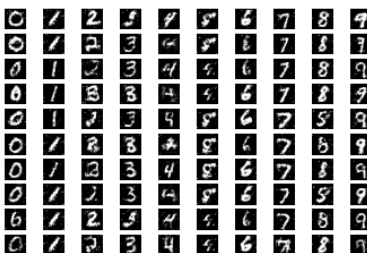
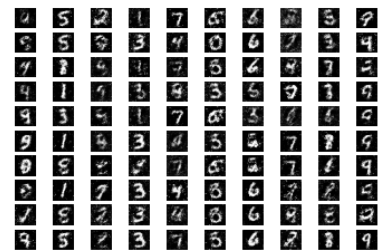
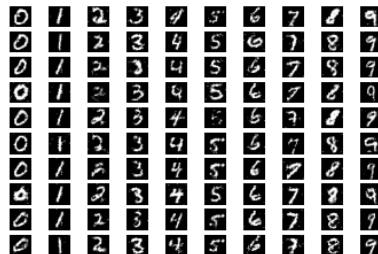
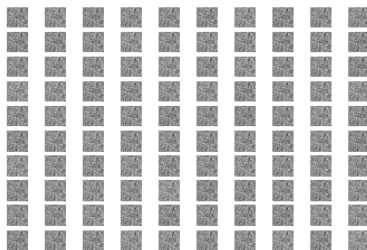
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	d	0	f	g	i	0	0	0	0
c	e	j	0	0	0	0	0	0	0
b S a 0									G

۲. این کد مربوط به آموزش یک GAN عادی با استفاده از مجموعه داده MNIST است. آن را به گونه‌ای ویرایش کنید که بتواند با استفاده از ساختار زیر به صورت شرطی داده تولید کند (از هر رقم که خواستیم تصویر تولید کند). پس از آموزش، از هر کلاس ۱۰ نمونه تولید کنید و نتایج را نمایش دهید (مجاز به استفاده از کدهای دیگر نیستید و لازم است همین کد را ویرایش کنید).



کد ضمیمه شده:

و تصاویر یک نمونه از اجرای آن را هم ضمیمه کردم:



که در مثال من خواسته ام که در ستون اول فقط ۰ را generate کند

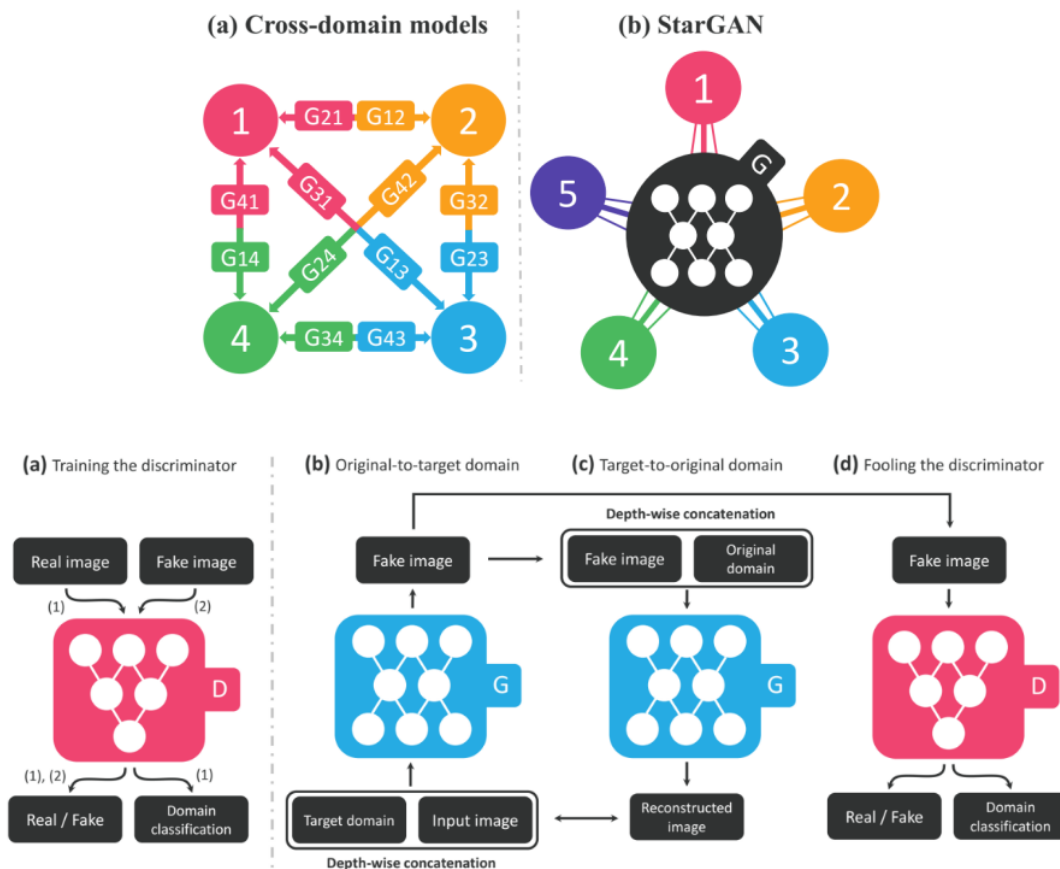
و در ستون دوم فقط ۱ و همین طور ادامه دادم

پس در هر شرط من تولید شماره ستون در هر خانه است.

می بینیم در طی ایپک های پشت سر هم توانست موفق شود.

۳. مقاله زیر را مطالعه کنید و به صورت مختصر و مفهومی دو شکل زیر را توضیح دهید.

Choi, Yunje, et al. "[Stargan: Unified generative adversarial networks for multi-domain image-to-image translation](#)." *IEEE conference on computer vision and pattern recognition*, 2018.



شکل اول:

در این مقاله هدف ایجاد مدلی است که بتواند هر ورودی ای در هر دامینی را به هر دامین دیگر در روشی **effective** تبدیل کند. فرض کنید شبکه ای داریم که چهره ی فرد را دریافت و ممکن است این چهره در حالت غمگین، شاد، پیر، تغییر جنسیت، ... و باشد

حال وظیفه شبکه تبدیل آن به چهره ای در یکی دیگر از حالت های فرد می باشد.

به طور نمونه : غمگین به شاد

در حالت GAN عادی ما نیاز به $k-1$ تا k generator هستیم که باید **train** کنیم اما در شبکه ی **StarGAN** هدف استفاده از فقط یک generator می باشد که با یک با آموزش بتواند هر تبدیل دو به دویی را انجام دهد. ایده آن هم ساده است و فقط علاوه بر تصویر باید اطلاعات **domain** آن را هم ورودی بگیریم و یک تبدیل قابل انعطاف را یاد بگیریم.

شکل دوم:

در این شکل سه قسمت داریم که در اشاره به یک نوع **ACGAN** دارد

در قسمت **a** با یک **discriminator** داریم که سعی دارد با گرفتن یک تصویر طبقه بندی کرده و **fake** یا **real** بودن آن را مشخص کند

در قسمت **b** ما یک generator داریم که با گرفتن یک تصویر ورودی (که معمولا رندوم است) و گرفتن اطلاعات دامنه اش یک تصویر **fake** در همان دامنه تولید کند

در قسمت **c** هم ما یک generator معکوس قسمت **b** داریم که سعی دارد با گرفتن اطلاعات دامنه و یک تصویر **fake** سعی دارد، تصویر اصلی که از روی آن این تصویر **fake** ساخته شده را بازسازی کند.

در قسمت **d** هم ما یک **discriminator** داریم که که دو کار انجام میدهد اولاً تشخیص تصویر **fake** و **real** بودن و ثانیاً دامنه ی تصویر را دسته بندی و تشخیص دهد. که در این صورت با دو بار آپدیت، وزن های بهتری خواهیم داشت