



دانشکده مهندسی برق و کامپیوتر

استاد درس: دکتر مهران صفایانی

دستورکار آزمایشگاه هوش محاسباتی

جلسه ۱

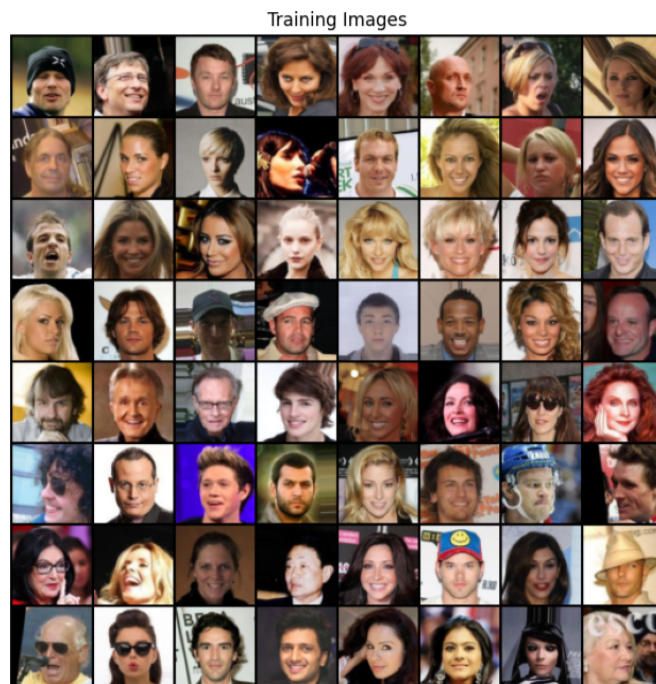
شبکه متخاصم مولد کانولوشنال عمیق

اهداف این جلسه

شما در این جلسه یاد خواهید گرفت که :

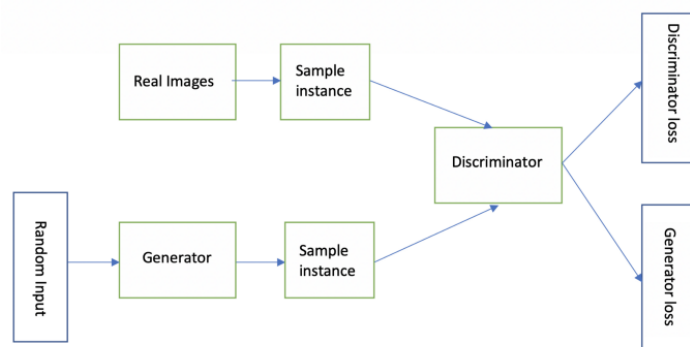
- مدل های GAN چیستند و مدل DCGAN را پیاده سازی کنید.
- با استفاده از TensorBoard خطا های مدل را نشان دهید.
- مشکل Vanishing Gradients و Mode Collapse را در مدل های GAN تشخیص دهید.

در این جلسه از دیتاست CelebA استفاده خواهیم کرد، این دیتاست شامل ۲۰۰ هزار عکس از شخصیت های معروف می باشد. در این تمرین قصد داریم تا مدلی پیاده سازی کنیم که عکس های شبیه به عکس های دیتاست تولید کند. نمونه ای از تصاویر موجود در دیتاست را در شکل زیر می توانید ببینید. این دیتاست را می توانید از این لینک دانلود کنید.



GAN چیست؟

در این شبکه ها دو مدل به طور همزمان آموزش داده میشوند که یکی از آن ها تولید کننده یا دیگری تفکیک کننده یا منتقد نام دارد. مدل تولید کننده آموزش داده میشود که تا جایی که ممکن هست عکس هایی نزدیک به واقعیت تولید کند و مدل منتقد آموزش داده میشود تا این عکس های جعلی را شناسایی و رد کند. اگر مدل تولید کننده بتواند مدل منتقد را فریب دهد در این صورت عکس هایی بسیار نزدیک به واقعیت تولید خواهد شد.

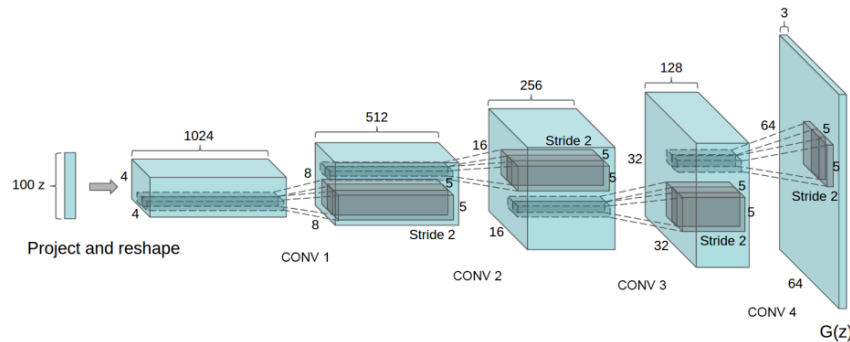


۱ مدل DCGAN چیست؟

مدل DCGAN یک توسعه مستقیم از GAN است که در بالا توضیح داده شد، با این تفاوت که به طور صریح از لایه‌های کانولوشنال و کانولوشن-ترانسپوز به ترتیب در تفکیک‌کننده و مولد استفاده می‌کند.

تمرین اول

در تمرین اول قصد داریم تا مدل مولد و تفکیک‌کننده را پیاده سازی کنیم، معماری مدل مولد را می‌توانید در شکل زیر ببینید.



۱. درباره لایه‌های کانولوشنال-ترانسپوز تحقیق کنید و آن‌ها را به مختصر توضیح دهید.

۲. مدلی با نام مولد (Generator) پیاده سازی کنید. این مدل شامل ۵ لایه کانولوشنال-ترانسپوز می‌باشد که بعد از هر لایه یک لایه BatchNorm قرار دارد و تابع فعال‌سازی هر لایه به جز لایه آخر ReLU می‌باشد. تابع فعال‌سازی لایه آخر \tanh می‌باشد.

- لایه اول: $\text{ConvTranspose}(\text{kernelSize} = 4, s = 1, p = 0, \text{bias} = \text{False}) \rightarrow \text{BN}() \rightarrow \text{ReLU}()$
- لایه دوم: $\text{ConvTranspose}(\text{kernelSize} = 4, s = 2, p = 1, \text{bias} = \text{False}) \rightarrow \text{BN}() \rightarrow \text{ReLU}()$
- لایه سوم: $\text{ConvTranspose}(\text{kernelSize} = 4, s = 2, p = 1, \text{bias} = \text{False}) \rightarrow \text{BN}() \rightarrow \text{ReLU}()$
- لایه چهارم: $\text{ConvTranspose}(\text{kernelSize} = 4, s = 2, p = 1, \text{bias} = \text{False}) \rightarrow \text{BN}() \rightarrow \text{ReLU}()$
- لایه پنجم: $\text{ConvTranspose}(\text{kernelSize} = 4, s = 2, p = 1, \text{bias} = \text{False}) \rightarrow \text{Tanh}()$

۳. مدلی با نام تفکیک‌کننده (Discriminator) پیاده سازی کنید. این مدل شامل ۵ لایه می‌باشد، در این مدل از لایه‌های کانولوشن، BatchNorm و تابع فعال‌سازی LeakyReLU و Sigmoid استفاده می‌شود. جزئیات مدل در زیر آمده است:

- لایه اول: $\text{Conv}(\text{kernelSize} = 4, s = 2, p = 1, \text{bias} = \text{False}) \rightarrow \text{LeakyReLU}(0.2)$
- لایه دوم: $\text{Conv}(\text{kernelSize} = 4, s = 2, p = 1, \text{bias} = \text{False}) \rightarrow \text{BN}() \rightarrow \text{LeakyReLU}(0.2)$
- لایه سوم: $\text{Conv}(\text{kernelSize} = 4, s = 2, p = 1, \text{bias} = \text{False}) \rightarrow \text{BN}() \rightarrow \text{LeakyReLU}(0.2)$
- لایه چهارم: $\text{Conv}(\text{kernelSize} = 4, s = 2, p = 1, \text{bias} = \text{False}) \rightarrow \text{BN}() \rightarrow \text{LeakyReLU}(0.2)$
- لایه پنجم: $\text{Conv}(\text{kernelSize} = 4, s = 1, \text{bias} = \text{False}) \rightarrow \text{Sigmoid}()$

۲ تابع خطا و آموزش مدل

در قسمت دوم این آزمایش می‌خواهیم مدل‌های پیاده سازی شده را آموزش دهیم، برای آموزش مدل از تابع خطای BCELoss استفاده می‌کنیم، این تابع به شکل زیر می‌باشد.

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^T, \quad l_n = -[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)]$$

از بهینه ساز Adam برای آموزش هر مدل استفاده می‌کنیم.

آموزش این مدل با آموزش مدل‌های دیگر کمی تفاوت دارد، ما نیاز داریم تا دو شبکه تفکیک‌کننده و مولد را آموزش دهیم. ازین رو آموزش این مدل به دو قسمت آموزش مولد و آموزش تفکیک‌کننده تقسیم می‌شود.

آموزش مدل تفکیک کننده

همانطور که می دانیم هدف تفکیک کننده این است که عکس های جعلی را از عکس های واقعی تفکیک کند. ازین رو آموزش این شبکه به دو قسمت تفکیک می شود.

۱. تصاویری از داده های واقعی به مدل داده و خطای مدل را بر روی آن ها به دست می آوریم و سپس مقدار گرادیان برای به روز رسانی پارامتر ها را به دست می آوریم و آن ها را به روز می کنیم.
۲. در قسمت دوم تصاویری جعلی با استفاده از شبکه مولد می سازیم و این تصاویر را به عنوان ورودی به مدل تفکیک کننده خود می دهیم. همانند مرحله اول مقدار خطا و گرادیان ها را محاسبه می کنیم و در نهایت پارامتر های مدل را به روز رسانی می کنیم.

آموزش مدل مولد

برای آموزش مولد همان داده های ساخته شده در قسمت قبل را به تفکیک کننده به روز رسانی شده می دهیم و خطا و گرادیان برای به روز رسانی پارامتر های شبکه مولد را محاسبه می کنیم و پس از آن مدل را به روز رسانی می کنیم.

تمرین دوم

۱. مرحله آموزش مدل را پیاده سازی کنید.
۲. میزان خطای هر شبکه را با استفاده از TensorBoard نمایش دهید.
۳. در مورد دو پدیده Mode Collapsing و Vanishing Gradients در شبکه ها متخاصم مولد تحقیق کنید و آن ها را به اختصار توضیح دهید.
۴. پارامتر های آزمایش را به گونه ای تغییر دهید که دو پدیده ذکر شده قابل مشاهده باشند.

نکات

- برای حل این آزمایش می توانید از فایل نوت بوکی که در کنار این پی دی اف می باشد استفاده کنید.

جمع بندی

در این آزمایش با شبکه های متخاصم مولد آشنا شدیم و با استفاده از آن ها چهره هایی نزدیک به چهره های دیتاست خود درست کردیم. در ادامه دو پدیده Mode Collapsing و Vanishing Gradients را مشاهده کردیم.