پرسش های طرح شده ـ یادگیری عمیق(دکتر فاطمی زاده) ـ بخش شبکه های مولد رقابتی (GAN)

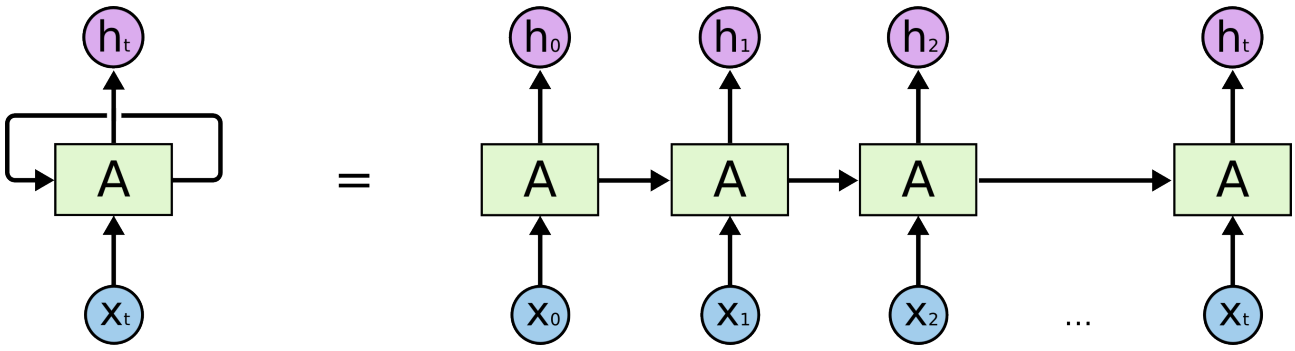
۱ Composer based on GAN!

در این تسک، همانند تمرین قبلی هدف پیاده سازی یک شبکه بازگشتی برای ابداع موسیقی بر اساس کد گذاری موسیقی های دلخواه است؛ ولی اینبار با این تفاوت که سناریو یاددهی مدل به صورت رقابتی (Adversarial) است.

* در لینک زیر،[۱] کلیات کار با فرمت midi و تنسور فلو و تبدیل این فرمت به ماتریس بیان شده است. به طور کلی فرمت mid. با پخش کننده های موسیقی قابل پخش است.

o داده های مناسب را از سایت دریافت کنید. دقت کنید در انتخاب موسیقی ها مختار هستید اما همانطور که احتمالا در تمرین قبلی دیدید؛ بهتر است موسیقی ها سبک و هارمونی مشابه داشته باشند. داده ها را برای ورودی دادن به شبکه پیش پردازش (تبدیل به فرمت تنسوری/ماتریسی) کنید.

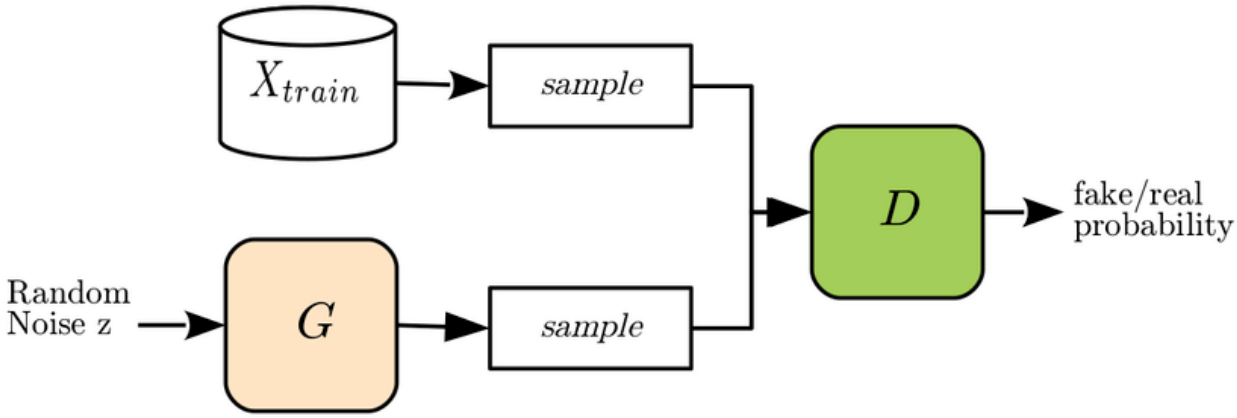
i شبکه مولد Generator را بر اساس کد گذاری خود طراحی کنید. ساختار کلی میتواند به صورت شکل باشد، اما در معماری آن به طور کلی سعی کنید ایده های خودرا پیاده سازی کنید. شبکه بهتر است بازگشتی باشد و از LSTM یا GRU در آن استفاده شود.



ii شبکه تشخیص دهنده Discriminator را بر اساس کد گذاری خود طراحی کنید. این شبکه باید یک آهنگ یا خروجی شبکه مولد را گرفته، و در نهایت یک خروجی بدهد که نمایانگر ساختگی بودن یا اصلی بودن این آهنگ است. میتوانید از همان معماری شبکه مولد برای این شبکه استفاده کنید؛ اما در خروجی فقط یک عدد بگیرید که توسط تابع فعالسازی ای همواره بین ۰ تا ۱ باشد.

iii در تمرین قبلی، سناریو به این صورت بود که همان آهنگ های دیتاست را به شبکه داده و سعی میشد شبکه بتواند آنها را بسازد؛ اما این بار ورودی شبکه مولد تنها یک بردار حالت تصادفی است. سناریو در نهایت به این صورت است؛ هر بار به ازای یک دسته بردار تصادفی، یک دسته آهنگ از G خروجی گرفته، به صورت یک Batch نگه دارید. سپس این داده هارا به همراه داده های دیتاست اصلی به D داده و خروجی را نگه دارید؛ شبکه ها حال باید بر اساس دو هزینه، بهینه شوند؛ D باید برای داده های اصلی X خروجی نزدیک ۱ بدهد و برای G(Z) خروجی نزدیک ۰. پس برای شبکه D، مساله به صورت یک

[۱] https://www.datacamp.com/community/tutorials/using-tensorflow-to-compose-music



شبکه ای در می آید که دو دسته داده با لیبل ۱ یا ۰ دارد. اما برای G، تابع هزینه به صورت برعکس بر روی خروجی فعلی D تعریف میشود:

$$Goal : min_G \ max_D \ V(D,G) = E_{x\sim data}[log(D(x))] + E_{z\sim random}[log(1 - D(G(z)))]$$

loss for D:

$$\sum_{i=1}^{K} -log(D(x_i)) + \sum_{i=1}^{l} -log(1 - D(G(z_i)))$$

loss for G:

$$\sum_{i=1}^{l} -log(D(G(z)))$$

حال باید شبکه ها بر اساس این هزینه ها جداگانه بهینه شوند، و کل این فرایند تکرار شود.

vi فرایند یاددهی را شروع کنید. برای این کار، یک حلقه اصلی بنویسید که در آن هر بار، G(z) ساخته میشود، سپس دیتا های اصلی و ساختگی با هم برای یاددهی D داده می شوند و لیبل آنها نیز به صورت متناظر اگر ساختگی است ۰ و اگر اصلی است ۱ قرار داده میشود. در مرحله بعدی، G بر اساس D تابع هزینه اش محاسبه شده و اپدیت می شود. در انتخاب تابع های هزینه مناسب، تعداد اپدیت D ،G در مرحله مختار هستید.