

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



شبکه‌های عصبی مصنوعی و یادگیری عمیق

مینی پروژه ۳ و تمرین ۴

دی ۱۴۰۰

فهرست سوالات

سوال ۱ – AC-GAN ۳

سوال ۲ – Deep Convolutional GAN ۴

سوال ۳ – WGAN ۷

سوال ۴ – maxnet&hammingnet ۱۱

سوال ۵ – SOM ۱۲

سوال ۱ – AC-GAN

در این بخش به بررسی شبکه AC-GAN^۱ می‌پردازیم.

این شبکه دارای قابلیت تولید تصاویر با درنظر گرفتن برجسب کلاس آن‌ها می‌باشد. مقاله مربوط به این شبکه را می‌توانید در [اینجا](#) مشاهده نمایید.

الف) ساز و کار شبکه AC-GAN را توضیح دهید.

ب) با استفاده از دیتاست ۱۰-Cifar شبکه را پیاده‌سازی نمایید. نمودارهای خطا را به همراه چند

نمونه تصویر تولید شده نمایش دهید.

ج) برای بررسی اثر بهینه‌ساز، شبکه را یک بار با Adam و بار دیگر با RMSprop با درنظر گرفتن

تعداد اپاک برابر آموزش دهید و نتایج این دو بهینه‌ساز را بررسی و گزارش نمایید.

(امتیازی): بجای ساختار generator در مقاله از شبکه U-Net استفاده نمایید و همانند بخش (ب)

شبکه AC-GAN را پیاده‌سازی کنید، نتایج را گزارش دهید.

* اگر در پیاده‌سازی و آموزش شبکه زمان بالا نیاز بود، تعداد اپاک کلی را به اندازه‌ای درنظر بگیرید که کیفیت تصاویر تولیدشده قابل تشخیص باشد و نیازی به کیفیت بالا نمی‌باشد.

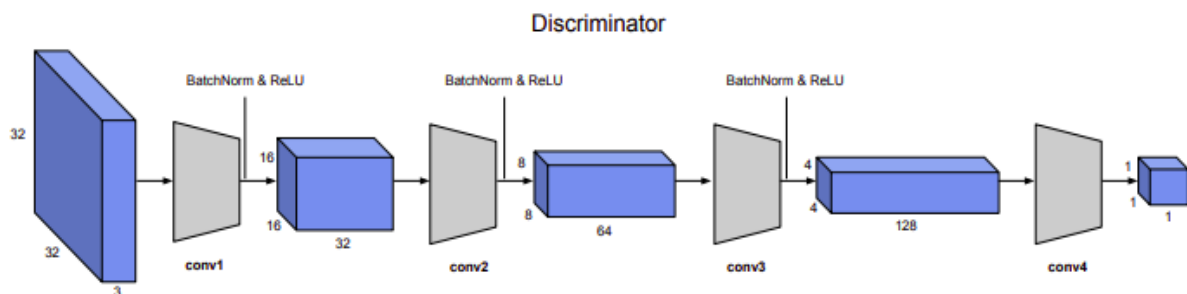
(*) برای درک بیشتر و آشنایی بهتر با شبکه GAN می‌توانید به این [لینک](#) مراجعه کنید که نحوه کار

این شبکه نمایش داده شده است.

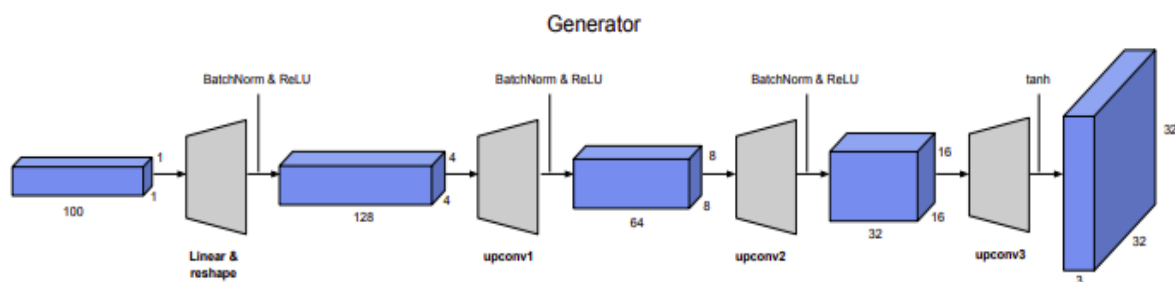
^۱ Auxiliary Classifier Generative Adversarial Network

سوال ۲ – Deep Convolutional GAN

در این سوال، هدف پیاده‌سازی یک Deep Convolutional GAN می‌باشد. این شبکه، یک شبکه ساده می‌باشد که از شبکه عصبی کانولوشنی برای بخش discriminator و از یک شبکه متشکل از transposed convolutions برای بخش generator استفاده می‌کند. برای پیاده‌سازی این شبکه، شما نیاز به مشخص کردن سه بخش دارید: بخش generator، بخش discriminator و بخش فرایند آموزش. دو شکل زیر معماری مربوط به بخش generator و discriminator شبکه DCGAN را نشان می‌دهد.



شکل ۱ معماری مربوط به بخش discriminator شبکه DCGAN



شکل ۲ معماری مربوط به بخش generator و شبکه DCGAN

سودو کد مربوط به فرایند آموزش این شبکه در جدول زیر گزارش شده است که به شما امکان تمرین ترجمه از ریاضیات به کد را فراهم می‌سازد.

Algorithm 1 GAN Training Loop Pseudocode

- 1: **procedure** TRAINGAN
- 2: Draw m training examples $\{x^{(1)}, \dots, x^{(m)}\}$ from the data distribution p_{data}
- 3: **Draw m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from the noise distribution p_z**
- 4: **Generate fake images from the noise: $G(z^{(i)})$ for $i \in \{1, \dots, m\}$**
- 5: **Compute the (least-squares) discriminator loss:**

$$J^{(D)} = \frac{1}{2m} \sum_{i=1}^m \left[\left(D(x^{(i)}) - 1 \right)^2 \right] + \frac{1}{2m} \sum_{i=1}^m \left[\left(D(G(z^{(i)})) \right)^2 \right]$$

- 6: Update the parameters of the discriminator
- 7: **Draw m new noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from the noise distribution p_z**
- 8: **Generate fake images from the noise: $G(z^{(i)})$ for $i \in \{1, \dots, m\}$**
- 9: **Compute the (least-squares) generator loss:**

$$J^{(G)} = \frac{1}{m} \sum_{i=1}^m \left[\left(D(G(z^{(i)})) - 1 \right)^2 \right]$$

- 10: Update the parameters of the generator
-

شکل ۳ سودو کد آموزش GAN

حال موارد زیر را بررسی کنید.

الف) پیاده‌سازی مربوط به یادگیری این نوع شبکه عصبی عمیق را انجام دهید. برای بررسی فرایند یادگیری و کیفیت تصاویر از یک دیتاست دلخواه که حاوی تصاویر emoji می‌باشد، استفاده کنید. (می‌توانید از دیتاست‌های موجود در <https://www.kaggle.com/subinium/emoji-full-emoji-dataset> استفاده کنید).

ب) چند راهکار برای پایدارسازی آموزش شبکه GAN توضیح دهید. همچنین، Gradient Penalty که یک مورد از این راهکارها می‌باشد را توضیح دهید. و ایده آن را به پیاده‌سازی خود اضافه کرده و فرایند آموزش را تکرار کنید. نتایج خود را با بخش قبلی مقایسه کنید. به طور مختصر، دلیل مفید بودن این ایده را شرح دهید.

ج) فرایند یادگیری را برای مقادیر مختلف hyperparameter موجود، انجام دهید. آیا اینها توانسته‌اند که فرایند یادگیری را پایدار کنند؟ چرا؟

د) همانطور که در بخش‌های قبل دیدیم، در معماری بخش generator شبکه DCGAN، از deconvolution استفاده می‌کردیم. در این قسمت، به جای استفاده از لایه‌های deconvolution، از لایه‌های upsampling استفاده کنید و فرایند آموزش را انجام دهید. نتایج را مقایسه کنید.

ه) ایده مقاله مربوط به Info GAN را به طور مختصر شرح دهید. این مورد را به پیاده‌سازی خود اضافه کرده و فرایند آموزش را انجام دهید. نتایج را مقایسه کنید.

سوال ۳ – WGAN

شبکه‌های GAN عادی دارای مشکلات متعددی هستند. و از این مشکلات می‌توان به ناپایداری شبکه، Vanishing Gradient و همین‌طور Mode Collapse اشاره کرد که مخصوصاً دو مورد آخر در اثر استفاده از تابع هزینه Min-Max رخ می‌دهند. پس از اثبات وجود چنین مشکلاتی در شبکه GAN اولیه کارهای بسیاری در جهت رفع این مشکلات ارائه شد که در ادامه شما را با چندین نوع از این راهکارها آشنا می‌کنیم.

I. جهت پایداری شبکه‌های GAN دوتا از ساده‌ترین کارها و اولین پیشنهاداتی که داده شد، استفاده از تکنیک‌های [One-sided label Smoothing](#) و همچنین [Add Noise](#) هست که می‌توان از آن‌ها استفاده کرد. ابتدا مقاله‌های مربوط به نحوه پیاده‌سازی این دو تکنیک را مشاهده و مطالعه کنید و سپس نحوه پیاده‌سازی و جزئیات هر کدام از این روش‌ها را بیان کرده و در نهایت با اصلاح شبکه DCGAN طراحی شده در سوال قبل، موارد خواسته شده در زیر را برای مجموعه داده‌های سوال قبل (و یا CIFAR-۱۰) و همچنین Fashion-MNIST ارائه دهید.

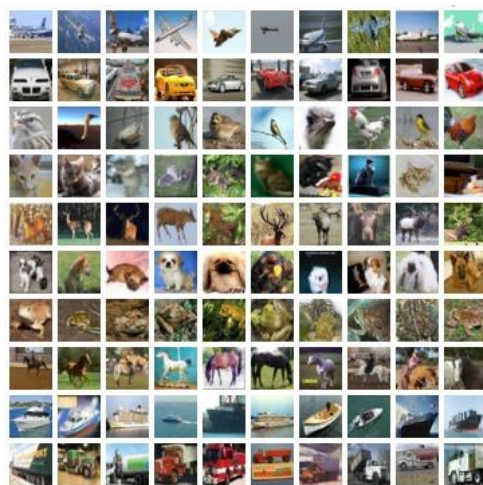
الف (توپولوژی لایه‌های مولد و تفکیک کننده

ب (نحوه ایجاد نویز

ج (نمودارهای Loss و Accuracy برای هر دو بخش Generator و Discriminator

د (نمودار نتایج شبکه مولد به ازای تعداد کافی از epoch روی داده‌های مسئله

توجه کنید که تصاویر تولیدی باید به صورتی مشابه شکل زیر باشند (یک ماتریس ۵ در ۵ تا ۱۰ در ۱۰)



شکل ۴: CIFAR-۱۰

II. در GAN های اولیه، یکی از اقداماتی که در جهت بهبود مشکل Vanishing Gradient و همچنین مشکل Mode-Collapse ارائه شد، تغییر تابع Loss و استفاده از Wasserstein Loss به جای Loss اولیه بود. این تابع هزینه جدید برای اولین بار در شبکه‌های [WGAN](#) مطرح شد. اکنون با مطالعه مقاله مربوطه ابتدا توضیح دهید که این Loss جدید چیست و تفاوت آن با Loss استفاده شده در شبکه‌های قبلی چیست. همچنین در ادامه ساختار WGAN را پیاده‌سازی کرده و موارد خواسته شده را برای مجموعه داده استفاده شده در سوال قبل (و یا ۱۰-CIFAR) ارائه دهید.

الف) توپولوژی لایه‌های مولد و تفکیک کننده

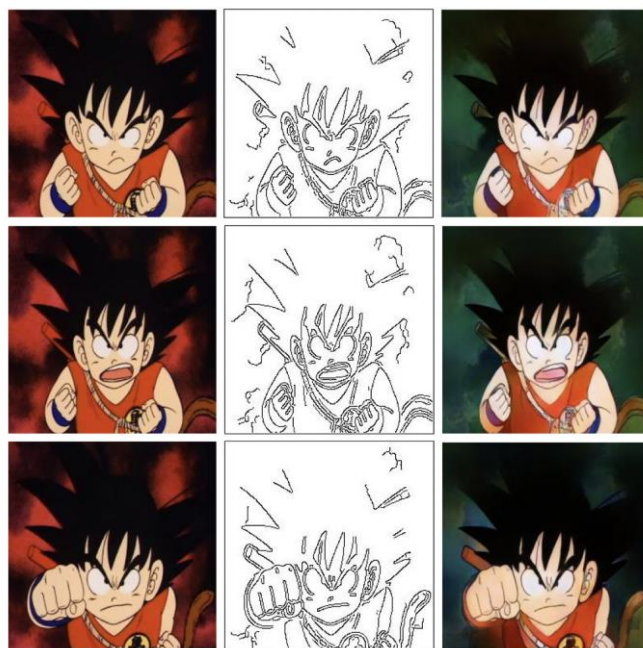
ب) نحوه ایجاد نویز

ج) نمودارهای Loss و Accuracy برای هر دو بخش Generator و Discriminator

د) نمودار نتایج شبکه مولد به ازای تعداد کافی از epoch روی داده‌های مسئله

III. (امتیازی) Image/Video Colorization

در این مسئله تمامی مراحل پیاده سازی یک شبکه برای رنگ آمیزی تصویر و ویدیو را انجام می‌دهیم. برای این کار می‌توانید از ایده‌های [مقاله پیشنهادی](#) نیز استفاده نمایید.



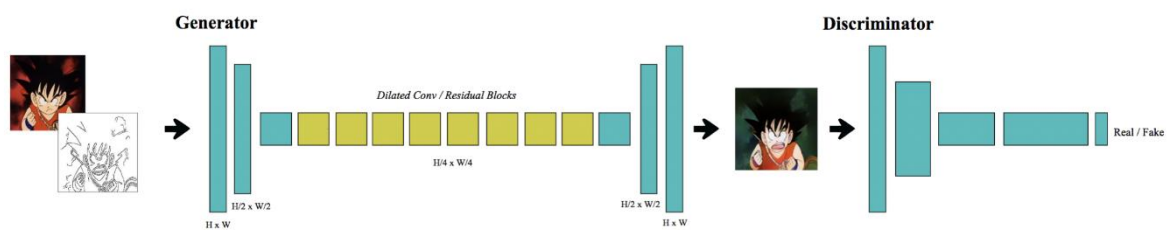
شکل ۵: انیمیشن مورد استفاده و تصاویری که باید استفاده کنید.

الف) از انیمیشن Dragonball را دانلود کرده و توسط OpenCV، در مجموع ۱۲۰ فریم از آن‌ها استخراج نمایید. ۹۰ عدد از این تصاویر را برای آموزش و ۳۰ تصویر دیگر را برای تست استفاده کنید. همچنین، از خود فریم‌ها به‌عنوان Ground Truth استفاده نمایید و تصاویر ورودی را در قسمت بعدی تهیه خواهید کرد.

ب) (Preprocessing) از آنجایی که ورودی شبکه مورد نظر تصاویر lineart خواهند بود باید مرزهای فریم‌ها را توسط یک الگوریتم edge detector استخراج کرده و از آن‌ها بعنوان ورودی استفاده نمایید. الگوریتم پیشنهادی برای این کار Canny edge detection با فیلتر Gaussian و انحراف معیار برابر با ۱ می‌باشد.

ج) (Colorization) حال یک شبکه Generative Adversarial بنویسید تا با گرفتن تصاویر ورودی، Ground Truth را تولید کند. شبکه را تست کرده، چهار نمونه از تصاویر تولید شده را به همراه Ground Truth آن‌ها در گزارش خود بیاورید.

راهنمایی: تابع هدف این قسمت تنها loss اصلاح شده GAN است.



شکل ۶: شبکه مورد استفاده

سوال ۴ - maxnet&hammingnet

۱- آرایه ی زیر را در نظر بگیرید:

$$[0.24, 0.3, 0.45, 0.57, 0.8, 0.69, 0.42, 0.26, 0.14]$$

الف) درخصوص نحوه ی عملکرد شبکه ی Mexican Hat توضیح مختصری ارائه دهید؛ سپس با استفاده از این شبکه ماکزیمم مقدار آرایه را برای $R_1 = 1$ و $R_2 = 4$ پیدا کنید. در هر تکرار، نمودار اندیس اعضای آرایه و سیگنال خروجی را رسم کنید. تابع فعال ساز را نیز بصورت زیر در نظر بگیرید:

$$f(x) = \begin{cases} 0 & x < 2 \\ x & 0 \leq x < 2 \\ 2 & x \geq 2 \end{cases}$$

ب) در چه شرایطی این شبکه مشابه یک شبکه ی MaxNet عمل می کند؟ مقادیر مناسب برای R_1 ، R_2 ، C_1 ، C_2 و t_{max} را در نظر بگیرید و قسمت الف را با این مقادیر تکرار کنید.

۲- شبکه ای طراحی کنید که بر اساس میزان شباهت هر بردار ورودی با بردارهای پایه ی زیر، آنها را دسته بندی کند، به نحوی که بردار ورودی در دسته بندی بردار پایه ای قرار بگیرد که به آن شبیه تر است.

$$e_1 = [1, -1, 1, -1, -1, -1]$$

$$e_2 = [-1, 1, -1, 1, 1, -1]$$

$$e_3 = [-1, -1, 1, 1, -1, 1]$$

$$e_4 = [1, 1, -1, 1, 1, -1]$$

الف) معماری شبکه را به طور کامل توضیح دهید.

ب) بردارهای زیر را بعنوان ورودی به شبکه اعمال کرده، با توجه به بردارهای پایه آنها را دسته بندی کنید و نتایج را در گزارش بیاورید.

$$v_1 = [-1, -1, 1, -1, 1, -1]$$

$$v_2 = [1, 1, 1, 1, -1, -1]$$

$$v_3 = [-1, -1, -1, 1, 1, -1]$$

$$v_4 = [1, -1, 1, 1, -1, 1]$$

$$v_5 = [1, 1, 1, -1, -1, -1]$$

$$v_6 = [1, -1, -1, 1, 1, 1]$$

$$v_7 = [-1, 1, -1, -1, -1, 1]$$

سوال ۵ - SOM

در این سوال می‌خواهیم به دسته‌بندی داده‌های دیتاست Fashion-MNIST بپردازیم. ابتدا یک شبکه SOM با ۲۲۵ نورون و با استفاده از ۱۰۰۰ داده اول دیتاست، آموزش داده و سپس نتایج خواسته شده را روی ۳۰۰ داده بعدی دیتاست گزارش دهید.

الف- نورون‌ها را روی نودهای یک شبکه 15×15 و با فرم مجاورت مربعی با شعاع $R = 1$ در نظر بگیرید.

ب- نمودار تعداد داده‌های هر خوشه (یک نمودار که محور x آن شماره خوشه و محور y تعداد داده‌های تست map شده به آن خوشه می‌باشد) را رسم کنید.

ج- روند تغییر خوشه‌بندی خود را تصویرسازی کنید. این کار را با تبدیل وزن‌های شبکه به تصاویر 420×420 پیکسل در ایپاک‌های مختلف انجام دهید. در واقع این تصویر شامل ۲۲۵ خانه 28×28 خواهد بود که هر خانه نمایانگر وزن‌های هر خوشه است. این کار را برای حداقل ۴ ایپاک انجام دهید تا روند به‌روزرسانی وزن‌ها به خوبی مشاهده شود.

د- تصویر وزن‌های خوشه‌های چگال را نشان دهید (حداکثر ۳۰ خوشه چگال).

نکات:

- مهلت تحویل این تمرین/مینی پروژه ۷ بهمن است.
- گزارش را در قالب تهیه شده که روی صفحه درس در Elearn بارگذاری شده، بنویسید.
- گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است. لطفاً تمامی نکات و فرض‌هایی که برای پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید را در گزارش ذکر کنید.
- در گزارش خود برای تصاویر زیرنویس و برای جداول هم بالانویس اضافه کنید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست. اما باید نتایج بدست آمده را گزارش و تحلیل کنید.
- دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند. بنابراین هرگونه نتیجه و یا تحلیلی که در شرح سوال از شما خواسته شده است را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- در صورت مشاهده تقلب امتیاز تمامی افراد شرکت‌کننده در آن، ۱۰۰- لحاظ می‌شود.
- برای انجام تمرین‌ها و مینی پروژه‌ها، تنها زبان برنامه نویسی مجاز Python است.
- استفاده از کدهای آماده برای تمرین‌ها به هیچ وجه مجاز نیست. اما برای مینی پروژه‌ها فقط برای قسمت‌هایی از کد و به عنوان راهنمایی برای پیاده‌سازی، می‌توانید از کدهای آماده استفاده کنید.
- نحوه محاسبه تاخیر به این شکل است: مهلت ارسال بدون جریمه تا تاریخ اعلام شده و با توجه به محدودیت زمانی این تمرین ارسال با تاخیر نخواهد داشت.
- لطفاً گزارش، فایل کدها و سایر ضمایم مورد نیاز را با فرمت زیر در سامانه مدیریت دروس بارگذاری نمایید.

HW#/PROJECT#_[Lastname]_[StudentNumber].zip

- در صورت وجود هرگونه ابهام یا مشکل می‌توانید از طریق رایانامه‌های زیر با دستیاران آموزشی مربوطه آقایا علی بارونی (سوال ۱)، سالار نوری (سوال ۲)، پرهام زیلوچیان مقدم (سوال ۳) احمد شعاع حقیقی (سوال ۵) و خانم کیمیا علوی (سوال ۴) در تماس باشید:

ali.barooni@ut.ac.ir

salar.nouri@ut.ac.ir

p.zilouchian@ut.ac.ir

kimia.alavi@ut.ac.ir

a.shoaahaghighi@gmail.com