به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین ششم

نام و نام خانوادگی	امیرحسین پورداود
شماره دانشجویی	۸۱۰۱۰۱۱۲۰
تاریخ ارسال گزارش	۱۴۰۱.۱۱.۷

فهرست

شکل‌ها

جدول‌ها

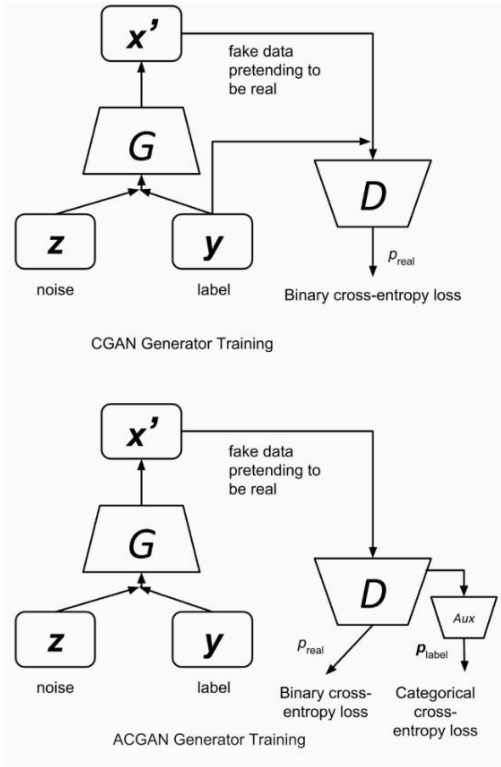
پاسخ ۲ – شبکه متخاصم مولد طبقه بندی کمکی و شبکه Wasserstein

۲-۱. شبکه متخاصم مولد طبقه بندی کمکی

در این بخش به بررسی شبکه AC-GAN میپردازیم. این شبکه نیز مشابه DCGAN دارای قابلیت تولید تصاویر البته با درنظر گرفتن برچسب کلاس آنها میباشد.

ابتدا CGAN و ACGAN را با هم مقایسه می کنیم. در هر دو CGAN و ACGAN، ورودی های ژنراتور، نویز و برچسب آن هستند. و خروجی آن، یک تصویر جعلی متعلق به برچسب کلاس ورودی است.

- ✓ برای CGAN، ورودی های تشخیص دهنده، یک تصویر (جعلی یا واقعی) و برچسب آن است و خروجی آن، احتمال واقعی بودن تصویر است.
- ✓ برای ACGAN، ورودی تشخیص دهنده یک تصویر است، در حالی که خروجی آن، احتمال واقعی بودن تصویر و برچسب کلاس آن است. شکل زیر تفاوت بین CGAN و ACGAN را در طول آموزش ژنراتور نشان می دهد:



شکل ۱ - تفاوت **CGAN** و **ACGAN**

۲-۱-۱-. پیاده سازی شبکه

مقدمه و تعریف

شبکه متخاصم مولد (GAN) یک معماری برای آموزش یک مدل مولد، معمولاً شبکه‌های عصبی پیچیده عمیق برای تولید تصویر است. این معماری هم از یک مدل مولد (Generator) تشکیل شده است که نقاط تصادفی را از یک فضای پنهان به عنوان ورودی می گیرد و تصاویر را تولید می کند و هم از یک تمایزکننده (Disciminator) برای طبقه بندی تصاویر به عنوان واقعی (از مجموعه داده) یا جعلی (تولید شده) تشکیل شده است. سپس هر دو مدل به طور همزمان در یک بازی حاصل جمع صفر آموزش داده می شوند.

یک Conditional GAN، به اختصار CGAN، توسعه‌ای از معماری GAN است که ساختار یا کلاس‌ها را به فضای پنهان اضافه می‌کند. آموزش مدل GAN به گونه‌ای تغییر می‌کند که ژنراتور هم با یک نقطه در فضای پنهان و هم یک برچسب کلاس به عنوان ورودی ارائه می‌شود و سعی می‌کند یک تصویر برای آن کلاس تولید کند. در ورودی تمایزکننده نیز هم، یک تصویر و یک برچسب کلاس ارائه می شود و باید مثل قبل واقعی یا جعلی بودن تصویر را طبقه بندی کند.

افزودن کلاس به‌عنوان ورودی، فرآیند تولید تصویر و فرآیند طبقه‌بندی تصویر را مشروط، به برچسب کلاس می‌کند، از این رو به آن CGAN گفته میشود. این اثر هم یک فرآیند آموزشی پایدارتر و هم یک مدل مولد برای تولید تصاویر از یک نوع خاص را بوجود می آورد.

Auxiliary Classifier GAN، یا به اختصار AC-GAN، گسترش دیگری از ساختمان معماری GAN بر روی شبکه CGAN است. توسط آگوستوس اودنا و همکاران معرفی شد. از Google Brain در مقاله ۲۰۱۶ با عنوان "سنتز تصویر شرطی با GAN های طبقه بندی کننده کمکی".

همانند GAN شرطی، مدل مولد در AC-GAN هم با یک نقطه در فضای پنهان و هم برچسب کلاس به عنوان ورودی ارائه می شود، به عنوان مثال. فرآیند تولید تصویر مشروط است. تفاوت اصلی، در مدل تشخیص‌گر است که بر خلاف Conditional GAN که تصویر و برچسب کلاس را به عنوان ورودی میگیرد، تنها تصویر به عنوان ورودی ارائه می شود. سپس مدل متمایز کننده باید واقعی یا جعلی بودن تصویر را مانند قبل پیش بینی کند و همچنین باید برچسب کلاس تصویر را پیش بینی کند.

معماری

معماری به گونه ای توصیف شده است که برای تشخیص دهنده و طبقه بندی کننده کمکی ممکن است مدل های جداگانه ای در نظر گرفته شوند که وزن مدل را به اشتراک می گذارند. اما در عمل، تفکیک کننده و طبقه بندی کمکی را می توان به عنوان یک مدل شبکه عصبی منفرد با دو خروجی پیاده سازی کرد.

- ✓ خروجی اول، یک احتمال واحد از طریق تابع فعال سازی سیگموئید است که "واقعیت یا جعلی" بودن تصویر ورودی را نشان می دهد و با استفاده از آنتروپی متقابل باینری (Binary Cross Entropy) مانند یک مدل متمایز کننده GAN معمولی بهینه شده است.
- ✓ خروجی دوم احتمال تعلق تصویر به هر کلاس از طریق تابع فعال‌سازی softmax است، مانند هر مدل شبکه عصبی طبقه‌بندی چند کلاسه، و با استفاده از آنتروپی متقابل دسته ای (Categorical Cross Entropy) بهینه شده است.

به طور خلاصه:

- ❖ مدل ژنراتور:

ورودی: نقطه تصادفی از فضای پنهان و برچسب کلاس.

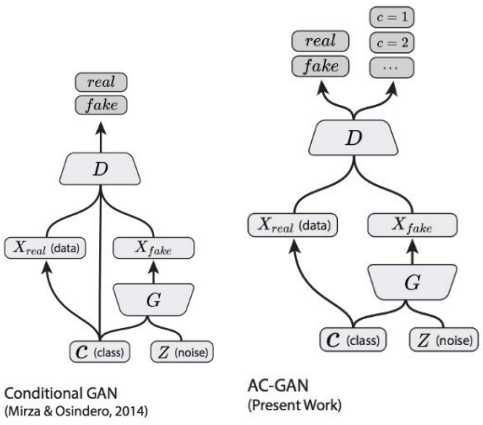
خروجی: تصویر تولید شده

- ❖ مدل تفکیک کننده:

ورودی: تصویر

خروجی: احتمال واقعی بودن تصویر ارائه شده و احتمال کلاس هر تصویر شناخته شده.

شکل زیر ورودی ها و خروجی های C-GAN و AC-GAN را خلاصه می کند و زمینه ای برای تفاوت ها فراهم می کند.



تمایز دهنده به دنبال به حداکثر رساندن احتمال طبقه بندی صحیح تصاویر واقعی و جعلی و پیش بینی صحیح برچسب کلاس یک تصویر واقعی یا جعلی است.

مولد به دنبال به حداقل رساندن توانایی تمایز کننده برای تشخیص تصاویر واقعی و جعلی است، در حالی که توانایی تشخیص دهنده را در پیش بینی برچسب کلاس تصاویر واقعی و جعلی به حداکثر می رساند.

مولد به دست آمده یک نمایش فضای پنهان را می آموزد که بر خلاف GAN شرطی مستقل از برچسب کلاس است. تأثیر تغییر GAN شرطی به این روش، هم منجر به فرآیند آموزشی پایدارتر است و هم توانایی مدل برای تولید تصاویر با کیفیت بالاتر با اندازه بزرگتر از آنچه قبلاً بود را ممکن میکند.

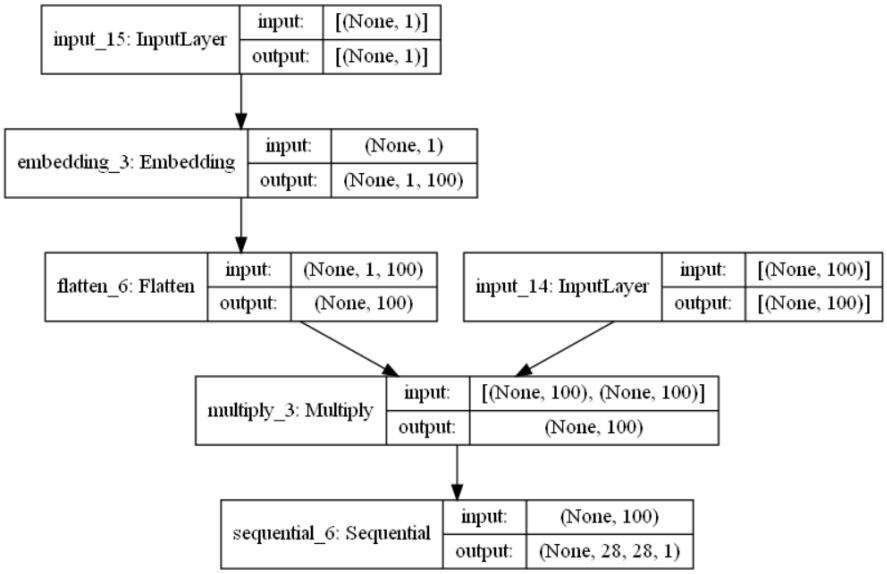
در مقاله دو مدل بهینه برای ساخت شبکه AC-GAN پیشنهاد کرده است که یکی بر مبنای دیتاست Imagenet برای تعداد کلاس ها و تعداد بالای تصاویر است و دیگری بر مبنای Fasion-MNIST است که تعداد کلاس های آن ۱۰ تا است و در زیر ساختار آن مشاهده میشود:

Operation	Kernel	Strides	Feature maps	BN?	Dropout	Nonlinearity
$G_x(z) - 110 \times 1 \times 1$ input						
Linear	N/A	N/A	384	×	0.0	ReLU
Transposed Convolution	5×5	2×2	192	√	0.0	ReLU
Transposed Convolution	5×5	2×2	96	√	0.0	ReLU
Transposed Convolution	5×5	2×2	3	×	0.0	Tanh
$D(x) - 32 \times 32 \times 3$ input						
Convolution	3×3	2×2	16	×	0.5	Leaky ReLU
Convolution	3×3	1×1	32	√	0.5	Leaky ReLU
Convolution	3×3	2×2	64	√	0.5	Leaky ReLU
Convolution	3×3	1×1	128	√	0.5	Leaky ReLU
Convolution	3×3	2×2	256	√	0.5	Leaky ReLU
Convolution	3×3	1×1	512	√	0.5	Leaky ReLU
Linear	N/A	N/A	11	×	0.0	Soft-Sigmoid
Generator Optimizer	Adam ($\alpha = [0.0001, 0.0002, 0.0003]$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)					
Discriminator Optimizer	Adam ($\alpha = [0.0001, 0.0002, 0.0003]$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)					
Batch size	100					
Iterations	50000					
Leaky ReLU slope	0.2					
Activation noise standard deviation	$[0, 0.1, 0.2]$					
Weight, bias initialization	Isotropic gaussian ($\mu = 0$, $\sigma = 0.02$), Constant(0)					

شکل ۲ - ساختار مولد و تمیزدهنده شبکه **ACGAN**

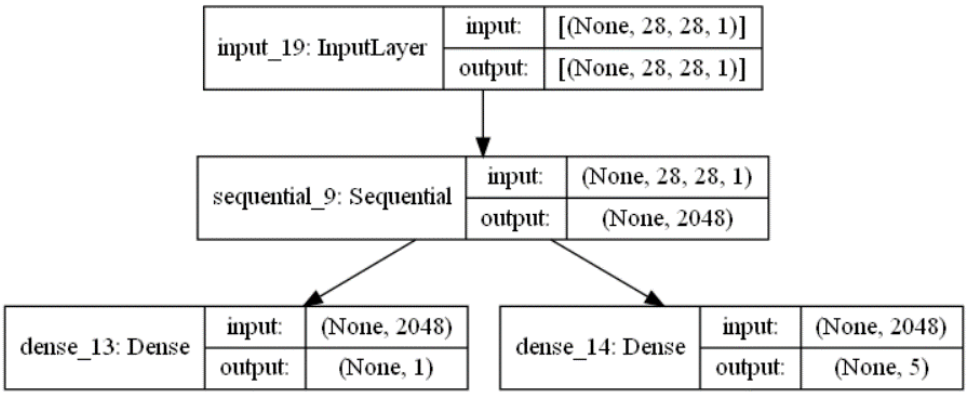
ما در این تمرین به دلیل کم بودن تعداد تصاویر دیتاست مورد نظر، از یک ساختار ساده تر استفاده کردیم.

مدل مولد:



شکل ۳- مدل مولد استفاده شده

مدل تمایزدهنده:



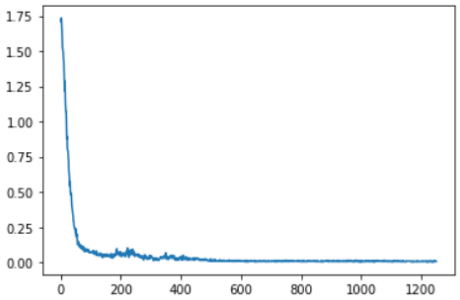
شکل ۴ - مدل متمایزگر استفاده شده

برای یک مرحله آموزشی معین، ابتدا مدل تفکیک کننده برای نیم دسته از نمونه های واقعی، سپس نیم دسته از نمونه های جعلی به روز می شود، که با هم یک دسته از به روز رسانی های وزن را تشکیل می دهند. سپس ژنراتور از طریق مدل ترکیبی GAN به روز می شود. نکته مهم این است که برچسب کلاس برای نمونه های جعلی روی ۱ یا واقعی تنظیم شده است. این تأثیر به روز رسانی ژنراتور به سمت بهتر شدن در تولید نمونه های واقعی در دسته بعدی را دارد.

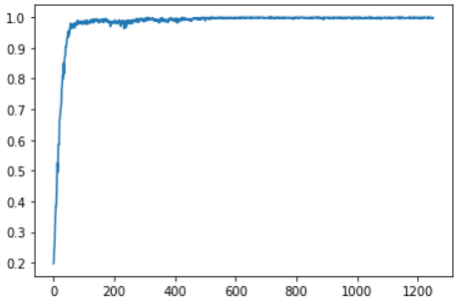
توجه داشته باشید، مدل تفکیک کننده و ترکیبی سه مقدار از دست دادن را از فراخوانی به تابع train_on_batch برمی‌گردانند. مقدار اول مجموع مقادیر تلفات است و می توان آن را نادیده گرفت، در حالی که مقدار دوم ضرر برای لایه خروجی واقعی/جعلی و مقدار سوم ضرر برای طبقه بندی برچسب لباس است.

۲-۱-۲. ارزیابی شبکه

نمودار loss:

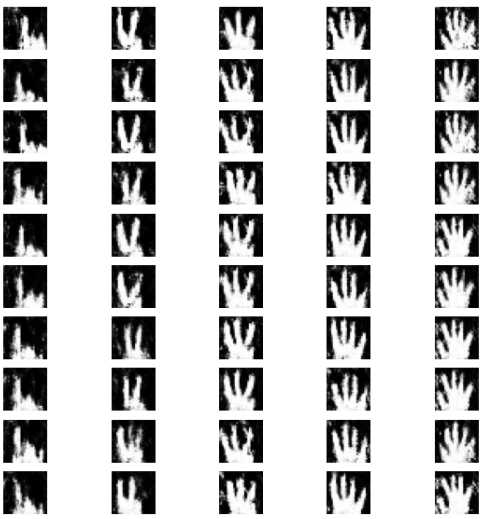


نمودار accuracy :



خروجی ها در هر مرحله نیز گرفته شده که پیشرفت مدل را نشان میدهد. و خروجی ها نیز بصورت GIF پشت هم قرار داده میشود و در کنار فایل گزارش ارسال میشود.

نمونه خروجی شبکه پس از ۱۲۵۰ epoch بصورت زیر میباشد:



شکل ۵ - نمونه خروجی مدل

۲-۲. شبکه متخاصم مولد Wasserstein

درGAN های اولیه، یکی از اقداماتی که در جهت بهبود مشکل Vanishing Gradient و همچنین مشکل Mode-Collapse ارائه شد، تغییر تابع Lossو استفاده از Wasserstein Loss به جای Loss اولیه بود. این تابع هزینه جدید برای اولین بار در شبکههای WGAN مطرح شد.

Wasserstein GAN یا به اختصار WGAN توسط Martin Arjovsky و همکارانش معرفی شد. در مقاله ۲۰۱۷ خود با عنوان "Wasserstein GAN". این یک توسعه از مدل GAN است که به دنبال یک روش جایگزین برای آموزش مدل مولد برای تقریب بهتر توزیع داده های مشاهده شده در یک مجموعه داده آموزشی است.

به جای استفاده از تشخیص دهنده برای طبقه بندی یا پیش بینی احتمال واقعی یا جعلی بودن تصاویر تولید شده، WGAN مدل تفکیک کننده را با منتقدی که واقعی بودن یا جعلی بودن یک تصویر را نمره می دهد، تغییر می دهد یا جایگزین می کند. انگیزه این تغییر یک استدلال نظری است که آموزش مولد باید به دنبال به حداقل رساندن فاصله بین توزیع داده های مشاهده شده در مجموعه داده آموزشی و توزیع مشاهده شده در نمونه های تولید شده باشد.

مزیت WGAN این است که فرآیند آموزش پایدارتر است و حساسیت کمتری نسبت به معماری مدل و انتخاب پیکربندی هایپرپارامتر دارد.

۲-۲-۱-. پیاده سازی شبکه

اجرای یک WGAN به چند تغییر جزئی در استاندارد Deep Convolutional GAN یا DCGAN نیاز دارد. تصویر زیر خلاصه ای از حلقه آموزشی اصلی برای آموزش WGAN را ارائه می دهد که از مقاله گرفته شده است:

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

تفاوت های پیاده سازی برای WGAN به شرح زیر است:

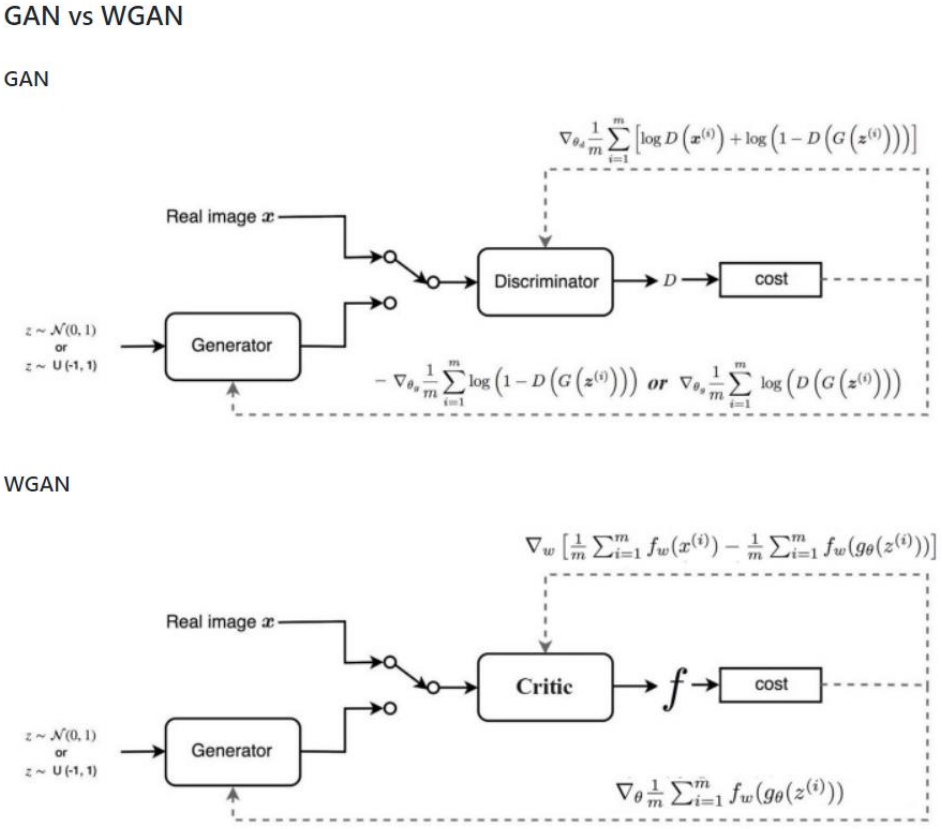
- ✓ از یک تابع فعال سازی خطی در لایه خروجی مدل انتقادی (به جای سیگموئید) استفاده شود.
  - ➢ DCGAN از تابع فعال سازی سیگموئید در لایه خروجی تشخیصگر برای پیش بینی احتمال واقعی بودن یک تصویر استفاده می کند. در WGAN، مدل انتقادی به یک فعال سازی خطی برای پیش بینی امتیاز "واقعیت" برای یک تصویر معین نیاز دارد.
- ✓ از برچسب -۱ برای تصاویر واقعی و ۱ برچسب برای تصاویر جعلی (به جای ۱ و ۰) استفاده شود.
  - ➢ DCGAN از کلاس ۰ برای تصاویر جعلی و کلاس ۱ برای تصاویر واقعی استفاده می کند و از این برچسب های کلاس برای آموزش GAN استفاده می شود. WGAN را می توان در جایی پیاده سازی کرد که از برچسب های کلاس -۱ برای تصاویر واقعی و برچسب های کلاس +۱ برای تصاویر جعلی یا تولید شده استفاده شود.
- ✓ **از ضرر Wasserstein برای آموزش مدل های منتقد و مولد استفاده شود.**
  - ➢ DCGAN تمایزگر را به عنوان یک مدل طبقه بندی باینری آموزش می دهد تا احتمال واقعی بودن یک تصویر داده شده را پیش بینی کند.
  - ➢ مدل WGAN استفاده از یک تابع ضرر جدید است که متمایز کننده را تشویق می کند تا امتیازی از واقعی یا جعلی بودن یک ورودی داده شده را پیش بینی کند. این امر نقش تمایزکننده را از یک طبقه‌بندی به یک منتقد برای امتیاز دادن به واقعی بودن یا جعلی بودن تصاویر تبدیل می‌کند، جایی که تفاوت بین امتیازات تا حد امکان زیاد است.
- ✓ وزن‌های مدل منتقد را پس از هر به‌روزرسانی دسته‌ای کوچک به یک محدوده محدود محدود شود (به عنوان مثال [-۰.۰۱، ۰.۰۱]).
  - ➢ DCGAN از برش گرادیان استفاده نمی کند، اگرچه WGAN برای مدل انتقادی نیاز به برش گرادیان دارد.
- ✓ در هر تکرار، مدل منتقد بیشتر از مولد به‌روزرسانی شود (مثلاً ۵).
  - ➢ در DCGAN، مولد و مدل تفکیک کننده باید به مقدار مساوی به روز شوند.
  - ➢ به طور خاص، تمایز با یک دسته از نمونه های واقعی و نیم دسته از نمونه های جعلی در هر تکرار به روز می شود، در حالی که ژنراتور با یک دسته از نمونه های تولید شده به روز می شود.

- ✓ از نسخه RMSProp شیب نزولی با نرخ یادگیری کم و بدون شتاب (به عنوان مثال ۰.۰۰۰۰۵) استفاده شود.
  - ➢ DCGAN از نسخه Adam نزول گرادیان تصادفی با نرخ یادگیری کم و حرکت متوسط استفاده می کند.
  - ➢ WGAN استفاده از RMSProp را با نرخ یادگیری کوچک ۰.۰۰۰۰۵ توصیه می کند.

از ضرر گرادیان **(Wasserstein)** برای آموزش این مدل استفاده شده است که بطور خلاصه ذکر شده است:

- DCGAN تمایزگر را به عنوان یک مدل طبقه بندی باینری آموزش می دهد تا احتمال واقعی بودن یک تصویر داده شده را پیش بینی کند.
- برای آموزش این مدل، تشخیص دهنده با استفاده از تابع تلفات متقابل آنتروپی باینری بهینه شده است. از همان تابع ضرر برای به روز رسانی مدل ژنراتور استفاده می شود.
- سهم اصلی مدل WGAN استفاده از یک تابع ضرر جدید است که تمایز کننده را تشویق می کند تا امتیازی از واقعی یا جعلی بودن یک ورودی داده شده را پیش بینی کند. این امر نقش تمایزکننده را از یک طبقه‌بندی به یک منتقد برای امتیاز دادن به واقعی بودن یا جعلی بودن تصاویر تبدیل می‌کند، جایی که تفاوت بین امتیازات تا حد امکان زیاد است.
- ما می توانیم از دست دادن Wasserstein را به عنوان یک تابع سفارشی در Keras پیاده سازی کنیم که میانگین امتیاز را برای تصاویر واقعی یا جعلی محاسبه می کند.
- امتیاز برای نمونه های واقعی به حداکثر می رسد و برای نمونه های جعلی به حداقل می رسد. با توجه به اینکه نزول گرادیان تصادفی یک الگوریتم کمینه‌سازی است، می‌توانیم برچسب کلاس را در امتیاز میانگین ضرب کنیم (مثلاً -۱ برای واقعی و ۱ برای جعلی که هیچ اثری ندارد)، که تضمین می‌کند از دست دادن برای تصاویر واقعی و جعلی به حداقل می‌رسد. شبکه.

در زیر تفاوت GAN و WGAN در تصویر مشاهده میشود:



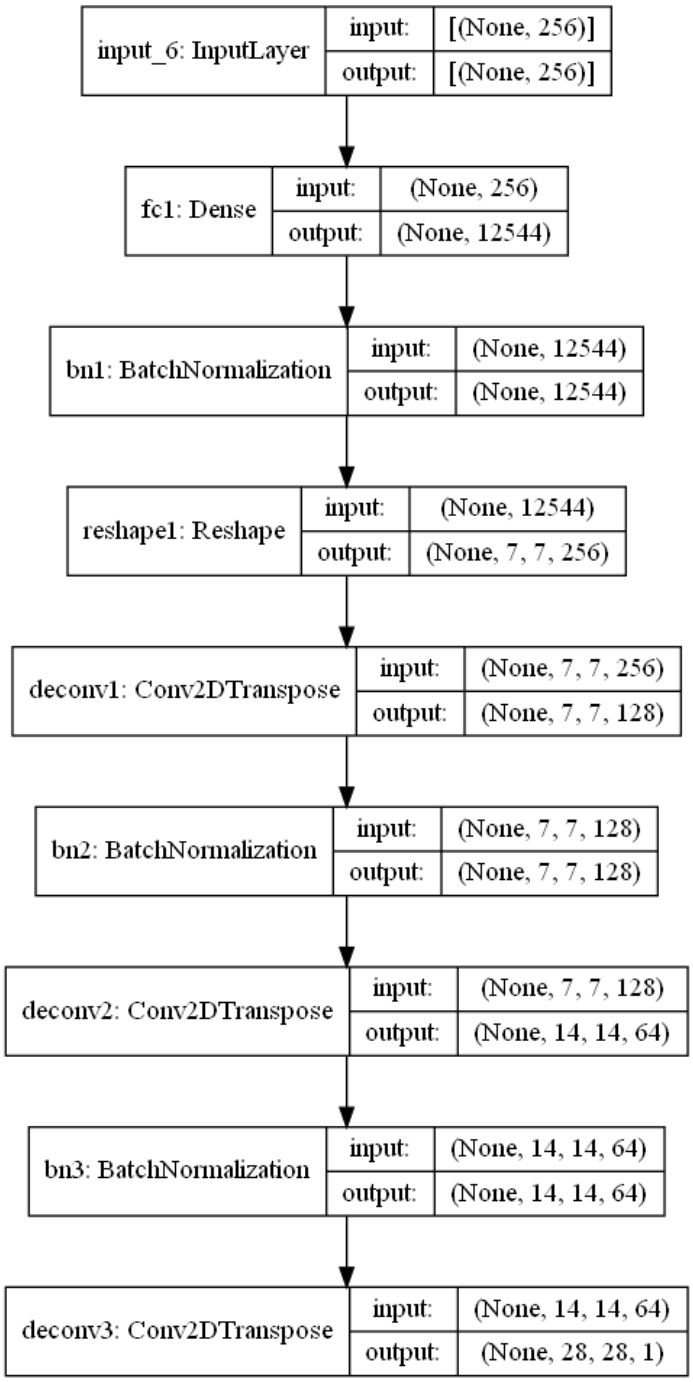
طراحی شبکه تقریباً یکسان است با این تفاوت که منتقد تابع سیگموئید خروجی ندارد. تفاوت عمده فقط در تابع هزینه است.

	Discriminator/Critic	Generator
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D(G(z^{(i)})))$
WGAN	$\nabla_w \frac{1}{m} \sum_{i=1}^{m} [f(x^{(i)}) - f(G(z^{(i)}))]$	$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f(G(z^{(i)}))$

با این حال، یک چیز عمده گم شده است. f باید یک تابع ۱-Lipschitz باشد. برای اعمال محدودیت، WGAN یک برش بسیار ساده برای محدود کردن حداکثر مقدار وزن در f اعمال می‌کند، یعنی وزن‌های تشخیص‌دهنده باید در محدوده خاصی باشند که توسط فراپارامترهای c کنترل می‌شود.

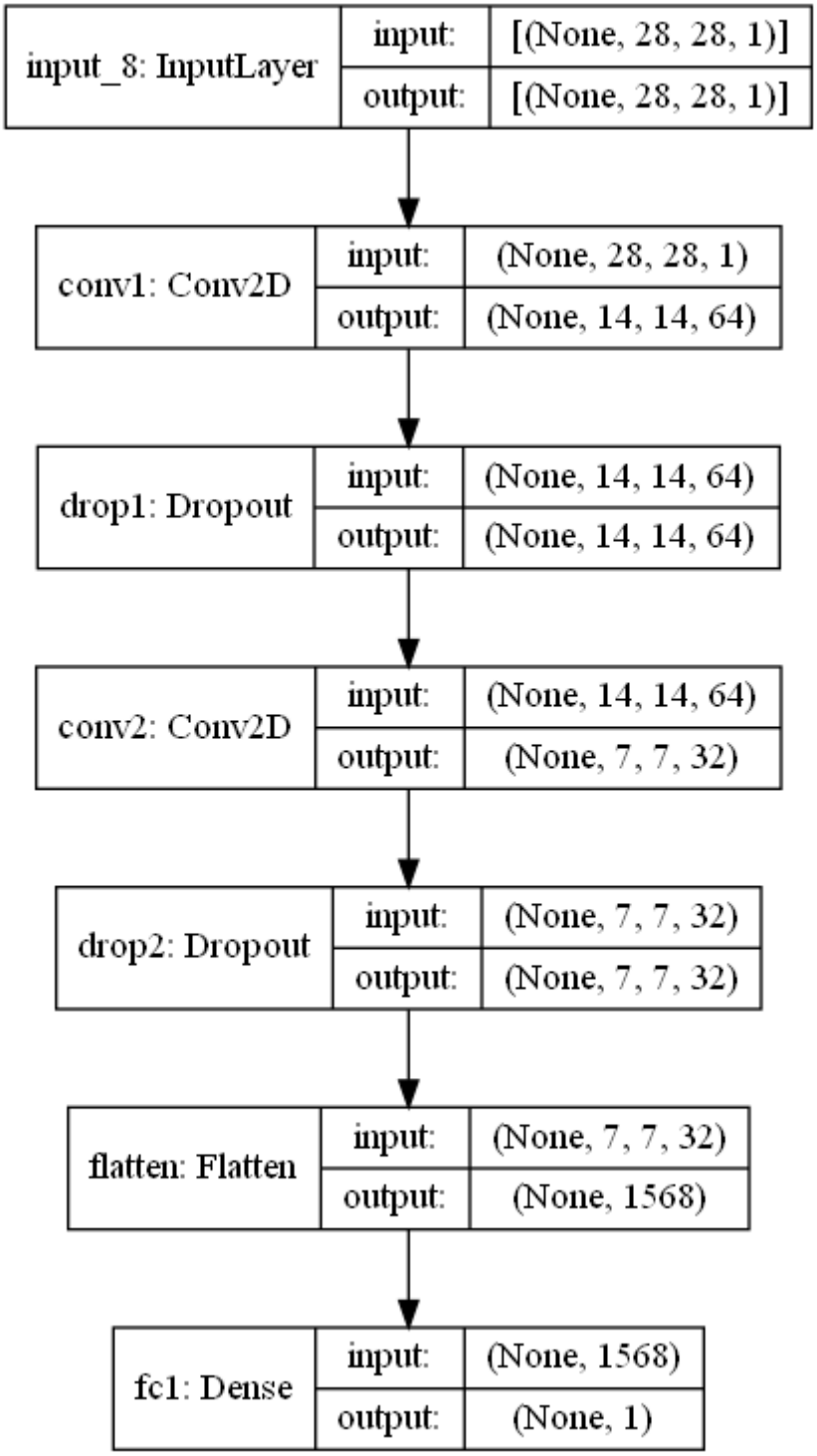
$$w \leftarrow w + \alpha \cdot \mathrm{RMSProp}(w, g_w)$$
$$w \leftarrow \mathrm{clip}(w, -c, c)$$

مدل مولد:



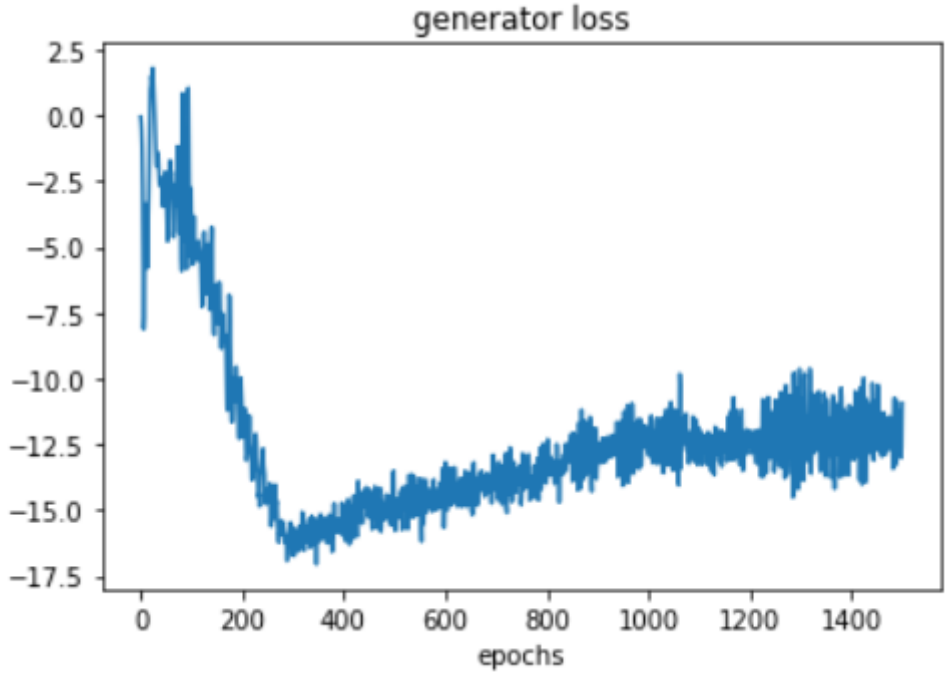
شکل ۶- مدل مولد استفاده شده

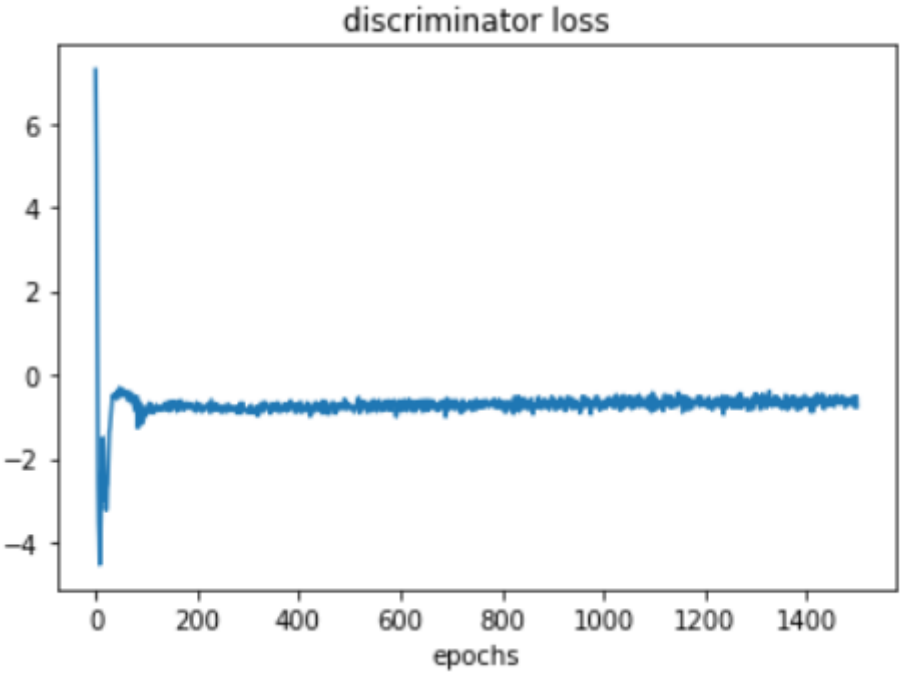
مدل تمایزدهنده:



شکل ۷ - مدل متمایزگر استفاده شده

۲-۲-۲. ارزیابی شبکه





نمونه خروجی شبکه پس از ۵۰۰۰ epoch بصورت زیر میباشد:

نسبت به شبکه عادی بع=هتر شده ولی در کل به دلیل کم بودن دیتا ست بصورت دقیق حاصل نمیشود.