DCGAN

① Architecture guidelines for stable DCGANs

(a) Do not use any pooling layers. Use strided CONV Layer in D and fractional-strided CONV Layer in G instead.

(b) Use batch norm in both G and D.

↳ Directly apply BN to all layers results in sample oscillation and instability.

↳ Do not apply BN to output layer of G and input layer of D

(c) Do not use any FC layers.

↳ can use global pooling instead which contribute to model stability but slow down the convergence.

(d) Use ReLU activation in G for all layers except for output layer which uses tanh (in order to project value to [-1, 1])

(e) Use Leaky ReLU in all layers of D

② Training Details (hyperparameter)

(1) No pre-processing. Only use tanh activation to scale the range of value into [-1, 1]

(2) SGD

(3) batch size 128

(4) All weighes were initialized from a zero-centered Normal distribution with standard deviation 0.02

(5) the slope of Leaky ReLU is 0.2

(6) Adam optimizer

(7) learning rate 0.0002

(8) momentum term β_1 0.5 instead of 0.9

③ Validate the Model Capacity. (衡量 Discriminator 在监督学习上的能力)

(1) 数据集 CIFAR

将 D 视作 feature extractor

方法: D of DCGAN pre-trained on Imagenet-1k.

use all of D's CONV layers

max pooling each layers representation to produce a 4×4 spatial grid

flaten and concatenate these grids to form a 28672 dimensional vector.

Put output of feature vector into L2-SVM to produce classification scores.

结果: Better than k-means, but not as good as Exemplar CNN

Is not trained on CIFAR! → domain robustness of learned feature

(2) 数据集 SVHN

同样作为 feature extractor

when Labeled data is scarce, DCGAN + L2-SVM outperform the previous works.

④ Visualization

→ 为了证明 DCGAN 是在学习 feature (representation)

而非简单的拟合/记忆图片.

(1) Walking in the latent space (隐空间 z)

Interpolation applyed to latent vector z results in smooth transition on generated image → features learned in z

(2) Visualizing D features

Use "guided backpropagation" to visualize the last CONV Layer of D. Significant board compared to baseline → feature learned in D

(3) Visualizing G1 features

① Use logistic regression to predict whether a feature activation is on a window or not based on 150 manually labeled generated image. Drop all activations greater than 0 (indicating a window). Applying the same vector z, G1 generates image without window! But also blurrier.

② Apply vector arithmetic for z, result in feature object combination or elimination

→ feature learned on G1.