Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

1 Citation

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

`https://arxiv.org/pdf/1511.06434v2.pdf`

2 Abstract

DCGAN is a Generative Adversarial Network (GAN) that uses Convolutional Neural Networks (CNN). Not only does it generate realistic images, it also learns general purpose features that are useful for classification.

3 Introduction

Let's use GANs to build a feature extractor for images. Our DCGAN combines GANs with CNNs and is pretty easy to train. It is a good feature extractor for images. We visualize the filters it learned.

4 Related Work

K-means is a popular unsupervised learning approach. Autoencoders are also quite good. Deep Belief Networks are also popular.

GANs are a new approach, but they are hard to train (the generator tends to keep generating the same image, images are noisy/blurry).

Using a deconvolution network or by doing gradient ascent on unit activation can help visualize what a filter has learned.

5 Approach and Model Architecture

Using GANs with CNNs is not easy. We tried a lot of model architectures. Our key insights are:

1. Replace pooling with strided convolution (generator) or fractional-strided convolution (discriminator)
2. Use batch normalization in generator and discriminator
3. Remove fully-connected hidden layers for deeper architectures

4. Use ReLU activation in generator for all layers except for the output, which uses tanh
5. Use LeakyReLU activation in the discriminator for all layers

6 Details of Adversarial Training

We train on Large-Scale Scene Understanding (LSUN) (pictures of bedrooms), ImageNet 1k, and a Faces dataset that we create. We scale pixels between $[-1, 1]$, initialize weights from zero-mean Gaussians, use a minibatch size of 128, and train with Adam.

Looking at generated images after each epoch, we confirm that our model is not just memorizing images.

To generate the Faces dataset, we take names of people and look them up on Wikipedia. We scrape the images and run OpenCV's face detector on them.

7 Empirical Validation of DCGAN's Capabilities

We train our GAN on ImageNet 1k. We then our the discriminator's convolutional features from all layers (max-pooled to create a 4×4 grid) as our feature representation. We then train an SVM on this representation and get great performance on CIFAR-10 classification. We also do well on Street View House Numbers (SVHN) classification.

8 Investigating and Visualizing the Internals of the Network

We don't do nearest-neighbor search or log-likelihood because those are terrible in image space.

We walk gradually through the latent space (i.e. gradually change the input noise to the generator) and verify that semantic changes to the generated images occur (the image starts to transform into another image) and that no sudden changes occur.

We identify the activations that are responsible for recognizing windows by labeling windows and fitting a logistic regression to identify which activations detect windows. Removing these activations then remove windows from the generated images.

We also summed latent vectors for two different training images and verified that the generated image was a blend of the two.

We also did gradient ascent on the image to see what maximized the activation, and we find that they activations respond to beds and other objects in bedrooms (the LSUN dataset is mainly bedrooms).

9 Conclusion and Future Work

DCGANs are easier to train than regular GANs, but they still sometimes collapse some filters into an oscillating mode.