Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Hongzhi Liu

Jun 24, 2018

Abstract

It is known to all that learning reusable feature representations from large unlabeled datasets has been an area of active research. Supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications in recent years. Comparatively, unsupervised learning with CNNs has received less attention. Today, I read a thesis written by Soumith Chintala, who is from Facebook AI Research. His team introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs) that have certain architectural constraints and demonstrate that they are a strong candidate for unsupervised learning. Training on various image datasets, they show convincing evidence that the deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator.

1. Overview of DCGANs

Unsupervised representation learning is a fairly well studied problem in general computer vision research as well as in the context of images. A classic approach to unsupervised representation learning is to do clustering on the data, and leverage the clusters for improved classification scores. Besides, Generative image models are well studied and fall into two categories: parametric and nonparametric.

The non-parametric models often do matching from a database of existing images, often matching patches of images, and have been used in texture synthesis, super-resolution and in-painting. Parametric models for generating images has been explored extensively. However, generating natural images of the real world have had not much success until recently. A variational sampling approach to generating images [7] has had some success but the samples often suffer from being blurry. A recurrent network approach [4] and a deconvolution network approach [3] have also recently had some success with generating natural images but they have not leveraged the generators for supervised tasks.

In this paper, Chintala and his team propose and evaluate a set of constraints on the architectural topology of Convolutional GANs that make them stable to train in most settings [8]. They name this class of architectures Deep Convolutional GANs (DCGAN). Besides, they use the trained discriminators for image classification tasks, showing competitive performance with other unsupervised algorithms. They then visualize the filters learnt by GANs and empirically show that specific filters have learned to draw specific objects. Furthermore, the team show that the generators have interesting vector arithmetic properties allowing for easy manipulation of many semantic qualities of generated samples. Training on various image datasets, the team show convincing evidence that the deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator.

2. Approach and Model Architecture

In this section, I will tell the architecture of Chintala. Core to their approach is adopting and modifying three recently demonstrated changes to CNN architectures.

2.1. Convolutional Net

The first is the all convolutional net [9] which replaces deterministic spatial pooling functions with strided convolutions, allowing the network to learn its own spatial downsampling. They use this approach in their generator, allowing it to learn its own spatial upsampling and discriminator.

2.2. Example Model Architecture

Second is the trend towards eliminating fully connected layers on top of convolutional features. The strongest example of this is global average pooling which has been utilized in state of the art image classification models. The team found global average pooling increased model stability but hurt convergence speed. A middle ground of directly connecting the highest convolutional features to the input

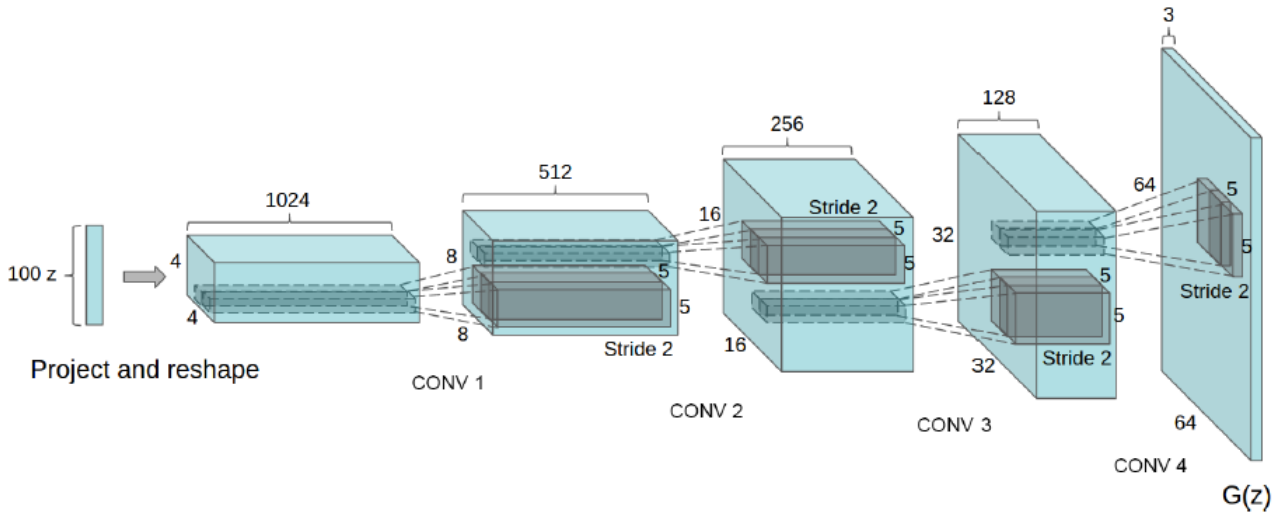

Figure 1. DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

and output respectively of the generator and discriminator worked well. The first layer of the GAN, which takes a uniform noise distribution Z as input, could be called fully connected as it is just a matrix multiplication but the result is reshaped into a 4-dimensional tensor and used as the start of the convolution stack. For the discriminator, the last convolution layer is flattened and then fed into a single sigmoid output. See Fig. 1 for a visualization of an example model architecture.

2.3. Batch Normalization

Third is Batch Normalization [6] which stabilizes learning by normalizing the input to each unit to have zero mean and unit variance. This helps deal with training problems that arise due to poor initialization and helps gradient flow in deeper models. This proved critical to get deep generators to begin learning, preventing the generator from collapsing all samples to a single point which is a common failure mode observed in GANs. Directly applying batchnorm to all layers but resulted in sample oscillation and model instability. This was avoided by not applying batchnorm to the generator output layer and the discriminator input layer.

2.4. Details Of Adversarial Training

Chintala trains DCGANs on three datasets, Large-scale Scene Understanding (LSUN) [10], Imagenet-1k and a newly assembled Faces dataset. As visual quality of samples from generative image models has improved, concerns of over-fitting and memorization of training samples have risen. To demonstrate how their model scales with more data and higher resolution generation, Chintala experiment a model on the LSUN bedrooms dataset containing a little over 3 million training examples. Recent analysis has shown that there is a direct link between how fast models learn and their generalization performance [5]. They show samples from one epoch of training as shown in Fig. 2, mimicking online learning, as an opportunity to demonstrate that their model is not producing high quality samples via simply overfitting or memorizing training examples.

3. Empirical Validation of DCGANs Capabilities

One common technique for evaluating the quality of unsupervised representation learning algorithms is to apply them as a feature extractor on supervised datasets and evaluate the performance of linear models fitted on top of these features.

On the CIFAR-10 dataset, a very strong baseline performance has been demonstrated from a well tuned single layer feature extraction pipeline utilizing K-means as a feature learning algorithm. When using a very large amount of feature maps (4800) this technique achieves 80.6% accuracy. An unsupervised multi-layered extension of the base algorithm reaches 82.0% accuracy [1].

Notably, the discriminator has many less feature maps compared to K-means based techniques but does result in a larger total feature vector size due to the many layers of 4×4 spatial locations. The performance of DCGANs is still less than that of Exemplar CNNs [2], a technique which trains normal discriminative CNNs in an unsupervised fashion to differentiate between specifically chosen, aggressively augmented, exemplar samples from the source dataset. Further

Figure 2. Generated bedrooms after one training pass through the dataset. Theoretically, the model could learn to memorize training examples but this is experimentally unlikely as they train with a small learning rate and minibatch SGD. The team are aware of no prior empirical evidence demonstrating memorization with SGD and a small learning rate.

Table 1. CIFAR-10 classification results using the pre-trained model. The DCGAN is not pretrained on CIFAR-10 but on Imagenet-1k and the features are used to classify CIFAR-10 images.

Model	Accuracy	Accuracy (400 per class)	max # of features units
Layer K-means	80.6%	63.7% (±0.7%)	4800
Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	80.6%	77.4% (±0.2%)	1024
DCGAN (theirs) + L2-SVM	82.8%	73.8% (±0.4%)	512

improvements could be made by finetuning the discriminator's representations, but they leave this for future work. Additionally, since their DCGAN was never trained on CIFAR-10 this experiment also demonstrates the domain robustness of the learned features.

References

[1] A. Coates and A. Y. Ng. Selecting receptive fields in deep networks. In *NIPS*, 2011. 2

[2] A. Dosovitskiy, P. Fischer, J. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE TPAMI*, 2014. 2

[3] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *CVPR*, 2015. 1

[4] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. DRAW: A recurrent neural network for image generation. *Computer Science*, 2015. 1

[5] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. *Mathematics*, 2015. 2

[6] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2

[7] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[8] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Computer Science*, 2015. 1

[9] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 1

[10] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *Computer Science*, 2015. 2