UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

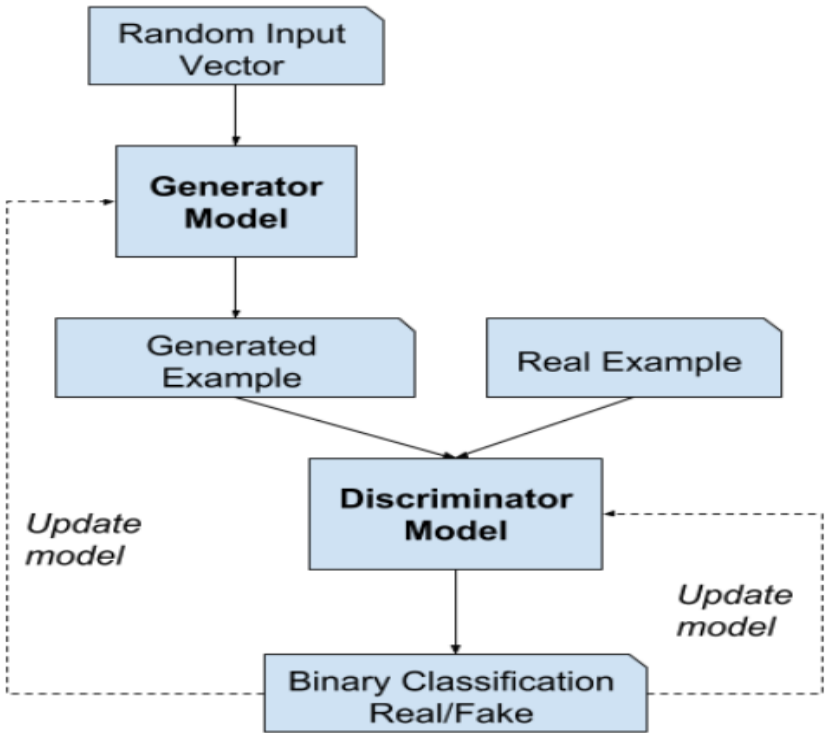
Author : Alec Radford & Luke Metz

Presented by : Dinesh Reddy

Apr 27th, 2021

Prerequisite

Generative Adversarial Network (GAN) are a deep learning based generative model. The GAN model architecture involves two sub-models: a generator model for generating new examples and a discriminator model for classifying whether generated examples are real (from the domain) or fake (generated by the generator model).



Problem Statement

In recent years, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. This paper help to bridge the gap between the success of CNNs for supervised learning and unsupervised learning. They 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 our deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator.

Introduction

The deep convolutional generative adversarial network, or DCGAN for short, is an extension of the GAN architecture for using deep convolutional neural networks for both the generator and discriminator models and configurations for the models and training that result in the stable training of a generator model.

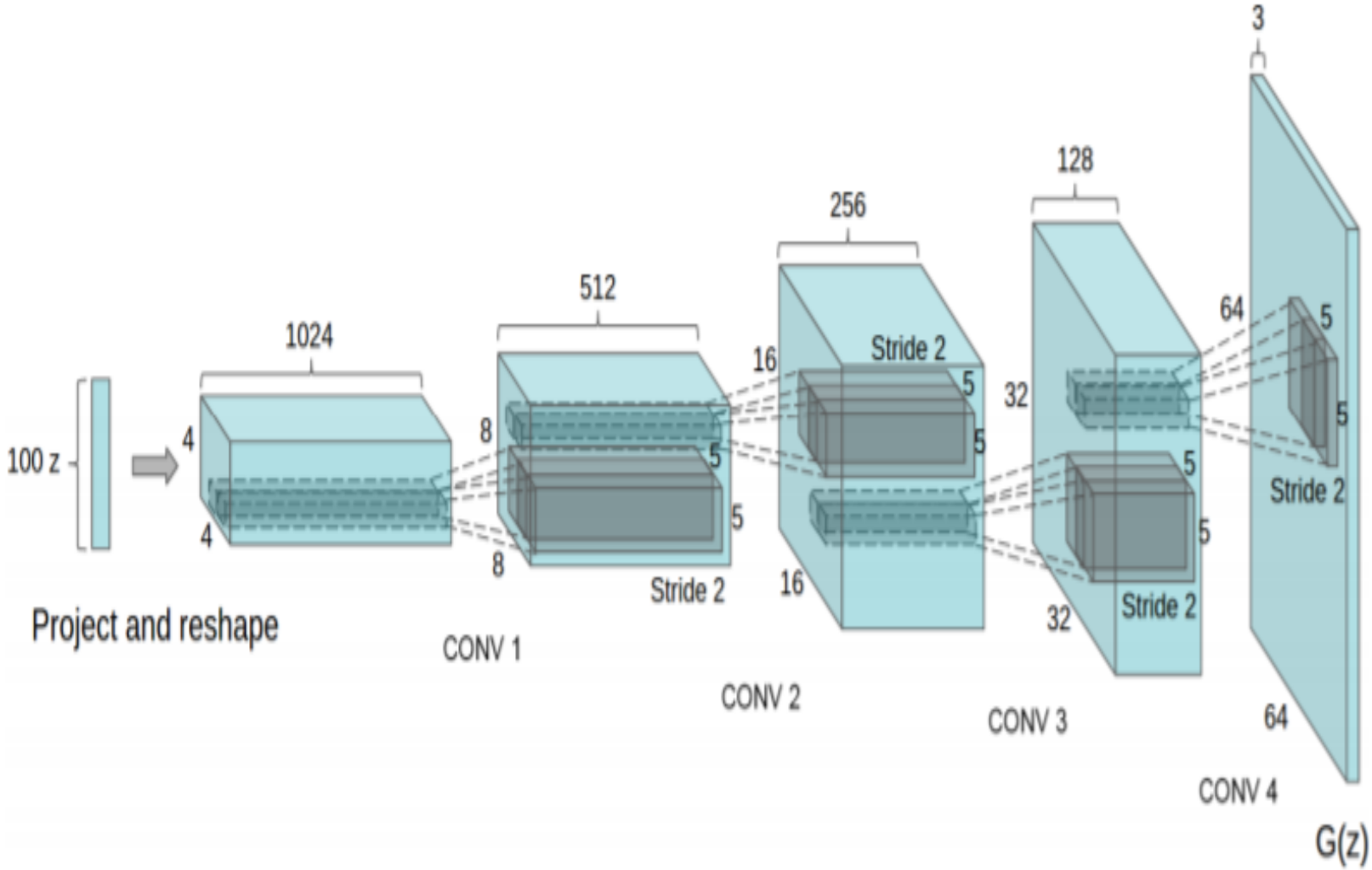
The core to the DCGAN architecture uses a standard CNN architecture on the discriminative model. For the generator, convolutions are replaced with upconvolutions, so the representation at each layer of the generator is actually successively larger, as it mapes from a low-dimensional latent vector onto a high-dimensional image.

Introduction (cont.,)

The architecture guidelines for stable Deep Convolutional GANs follows as below:

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architecture.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

Architecture

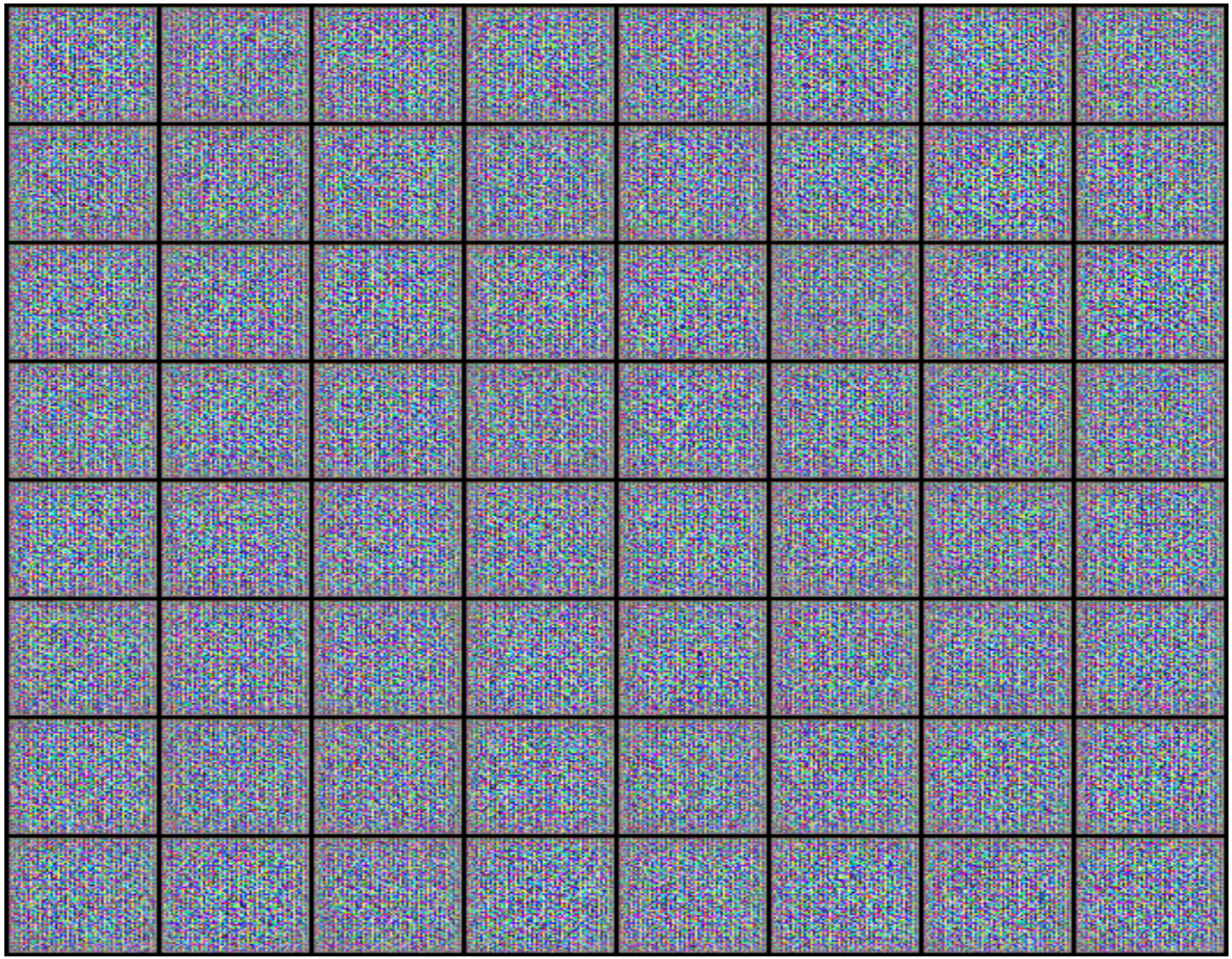


Implementation

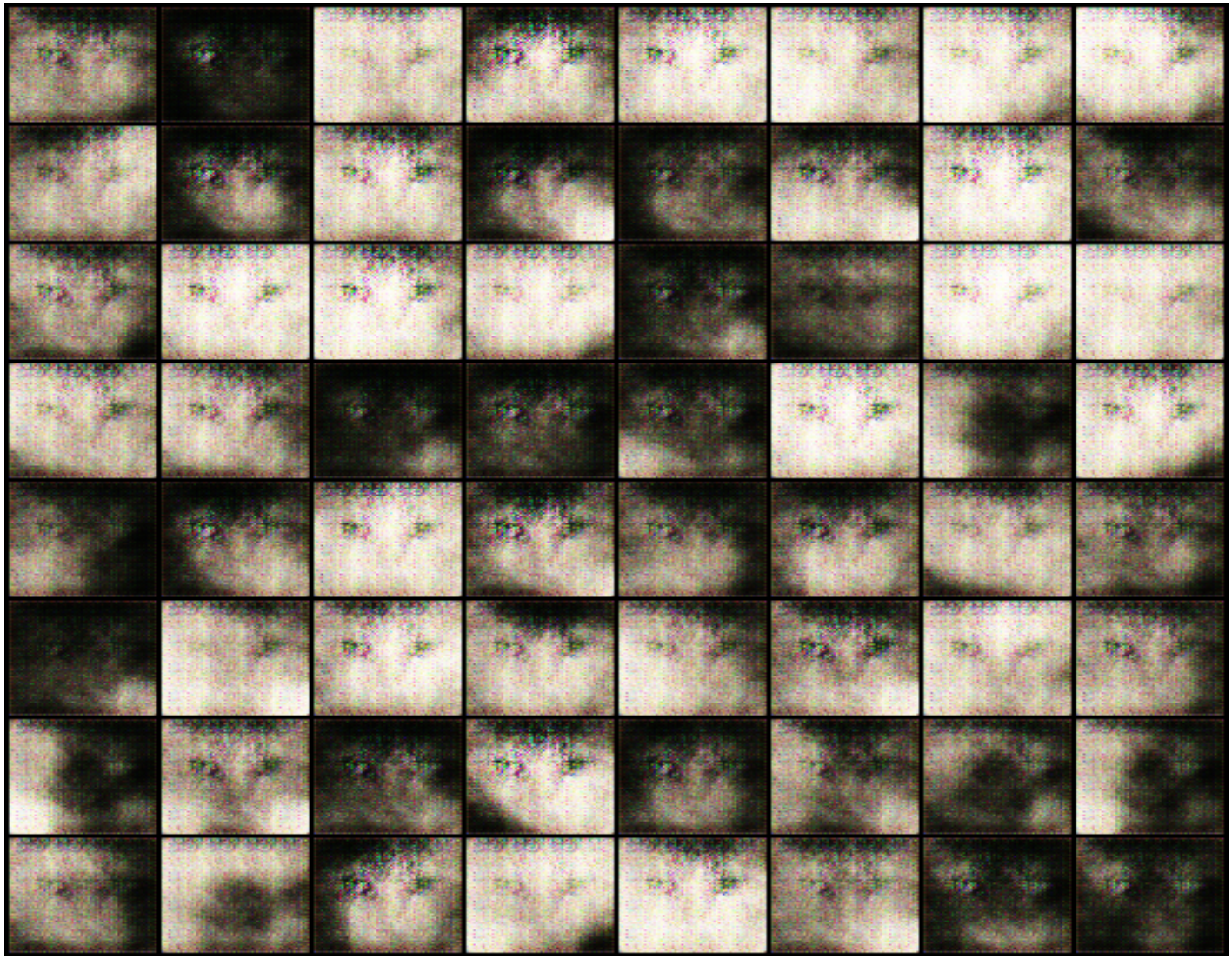
Here I have implemented DCGAN by training images of Cats Faces to generate new cats faces. Cats Faces Dataset (Available on Kaggle) which consists of more than 15,700 cats images. Sample images of cats faces are shown below.

Real Images from the dataset

Results



Results (cont.,)



Results (cont.,)

Generated Image

THANK YOU