Unsupervised Representation Learning Through Deep Convolutional Generative Adversarial Networks (DCGANs)

-Alec Radford & Luke Metz
indico Research
Boston, MA
-Soumith Chintala
Facebook AI Research
New York, NY

Presented by
Nikunj Gupta

Contributions

- Stabilizing the training of GANs in most settings.
  - By proposing and evaluating a set of constraints on the architecture of GANs.
- Usage of trained discriminators for image classification tasks.
- Visualization of the filters learnt by GANs
  - And empirically show that specific filters have learned to draw specific objects.
- Showing that generators have interesting vector arithmetic properties
  - For easy manipulation of many semantic qualities of generated samples.

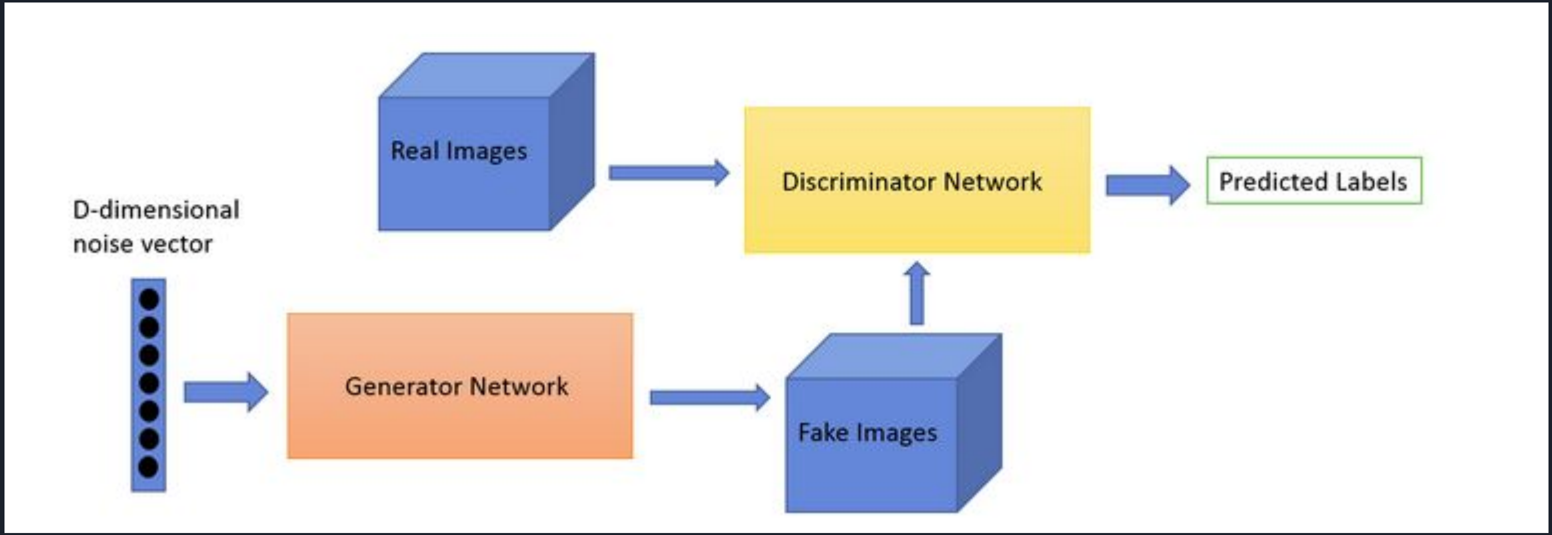
Related Work

- Representation Learning from Unlabelled Data
  - Clustering (K-means, etc.)
  - Hierarchical Clustering
  - Auto encoders
  - Deep Belief Networks

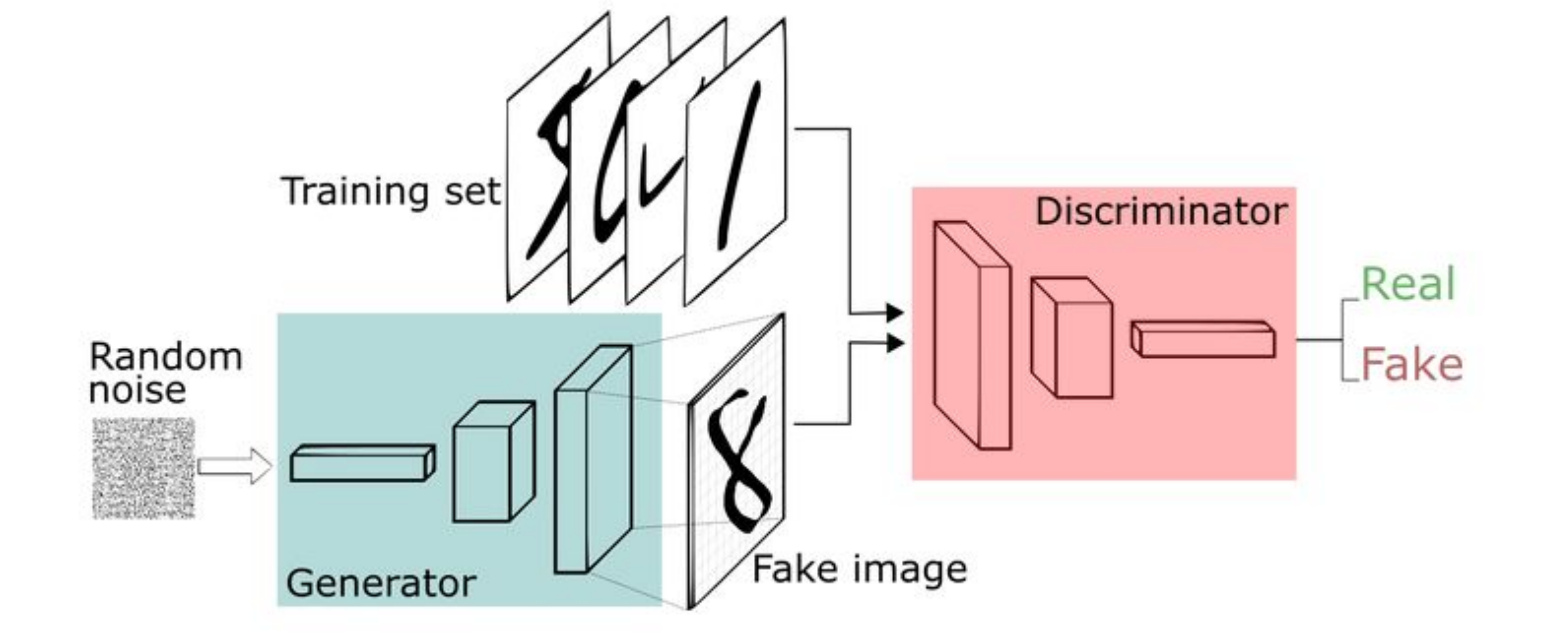
- Generating Natural Images
  - Variational Sampling Approach in GANs
  - Laplacian Pyramid Extension to GANs

- Visualizing the Internals of CNN
  - Using deconvolutions and filtering the maximal activations (Zieler et. al.)

GAN Architecture



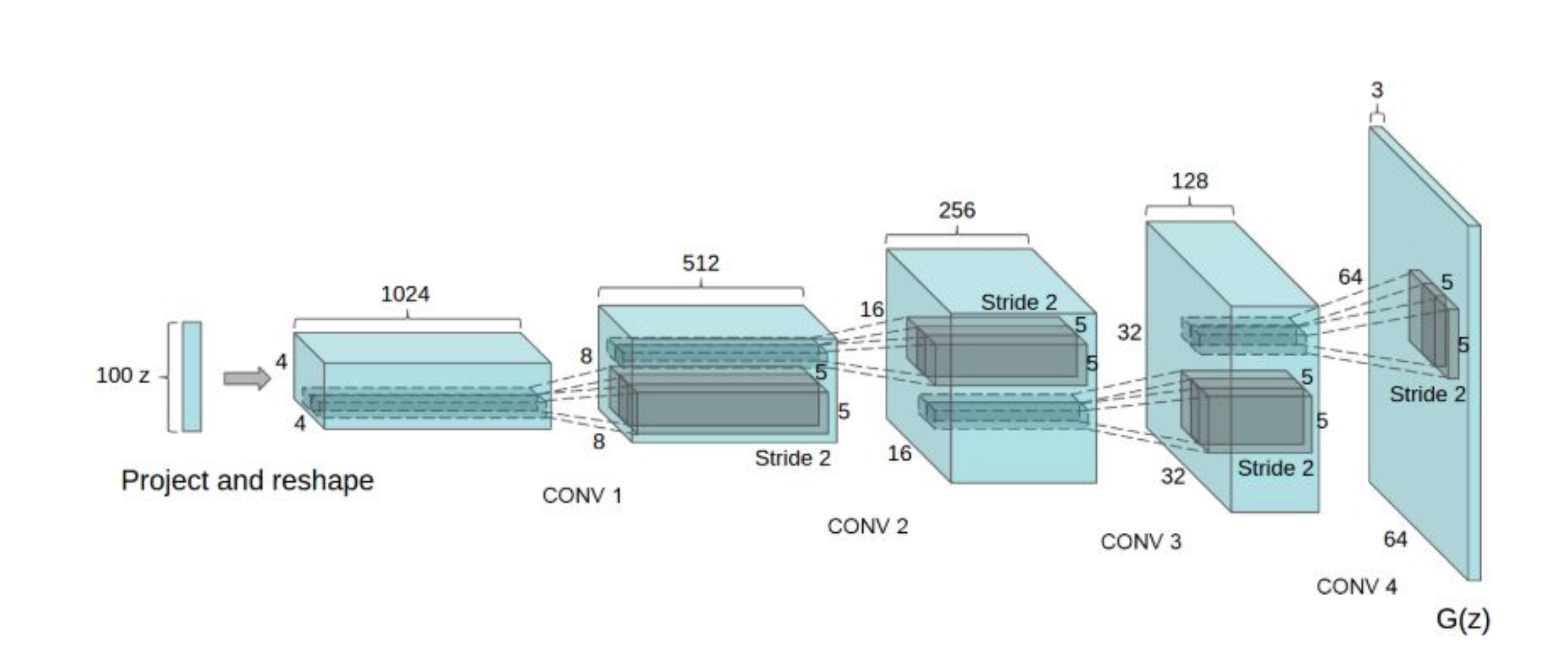
Inside GAN Architecture



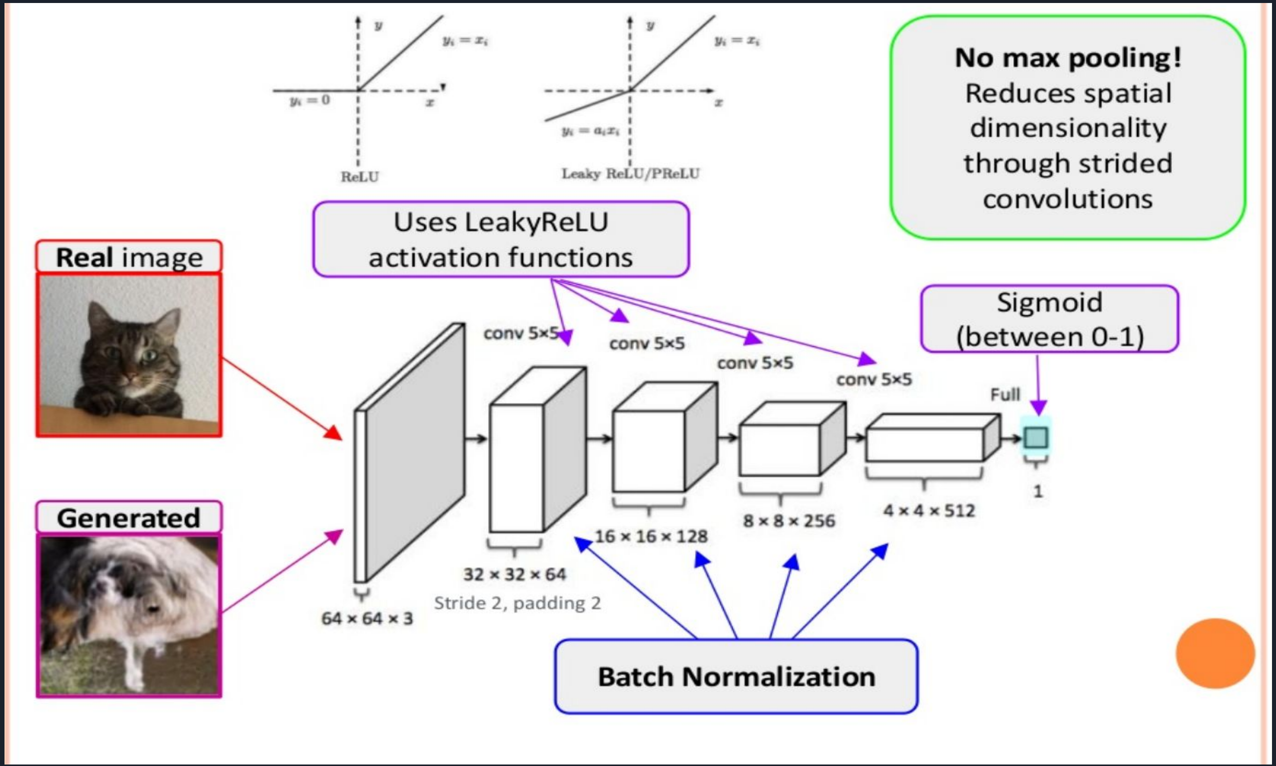
Approach and Model Architecture

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

Generator Network



Discriminator Network



Goals

- Generator Goal: $D(G(z)) = 1$
- Discriminator Goal: $D(G(z)) = 0$,

- Both goals are conflicting and unsupervised.

Details of Adversarial Training

- The training is done on 3 datasets: LSUN, ImageNet-1K, Faces Dataset
- Preprocessing: Scale images between -1 and +1 (tanh range)
- Mini batch SGD (m = 128)
- Weight Initialisation: zero centered normal distribution (std dev = 0.02)
- Leaky ReLU slope: 0.2
- Adam Optimizer
- Learning Rate: 0.0002
- Momentum Term = 0.5, to stabilize training

Results

Classification using the Discriminator Network

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

Model	Accuracy
1 Layer K-means	80.6%
3 Layer K-means Learned RF	82.0%
View Invariant K-means	81.9%
Exemplar CNN	84.3%
DCGAN (ours) + L2-SVM	82.8%

Table 2: SVHN classification with 1000 labels

Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWWAE without dropout	27.83%
SWWAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

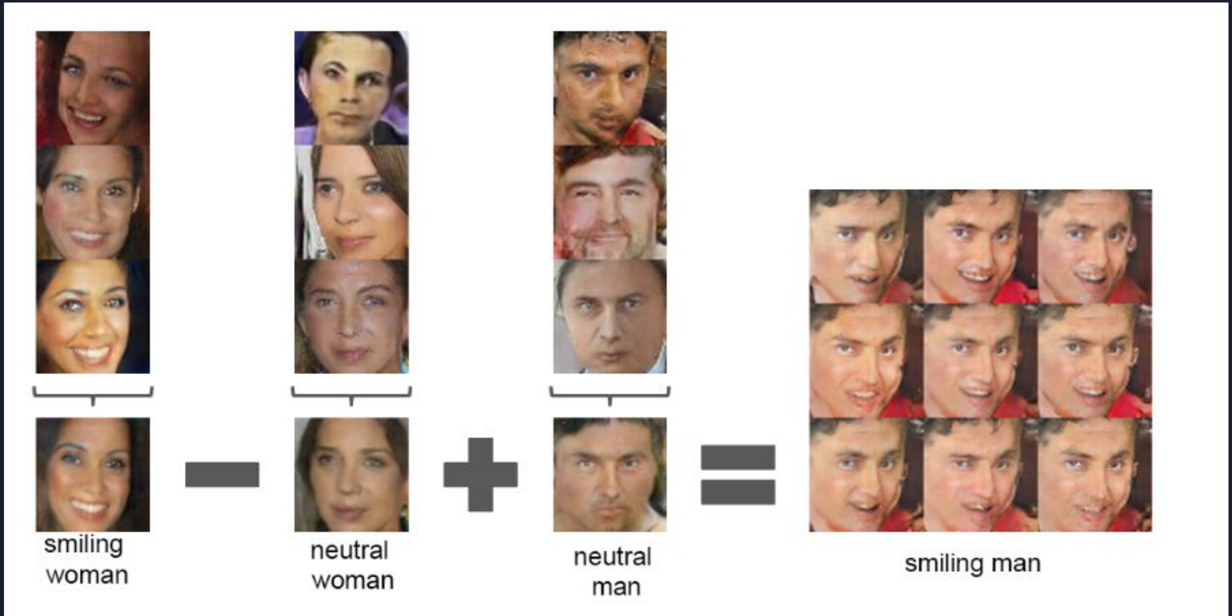
Bedroom Generation from LSUN Dataset

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 we train with a small learning rate and minibatch SGD. We are aware of no prior empirical evidence demonstrating memorization with SGD and a small learning rate.

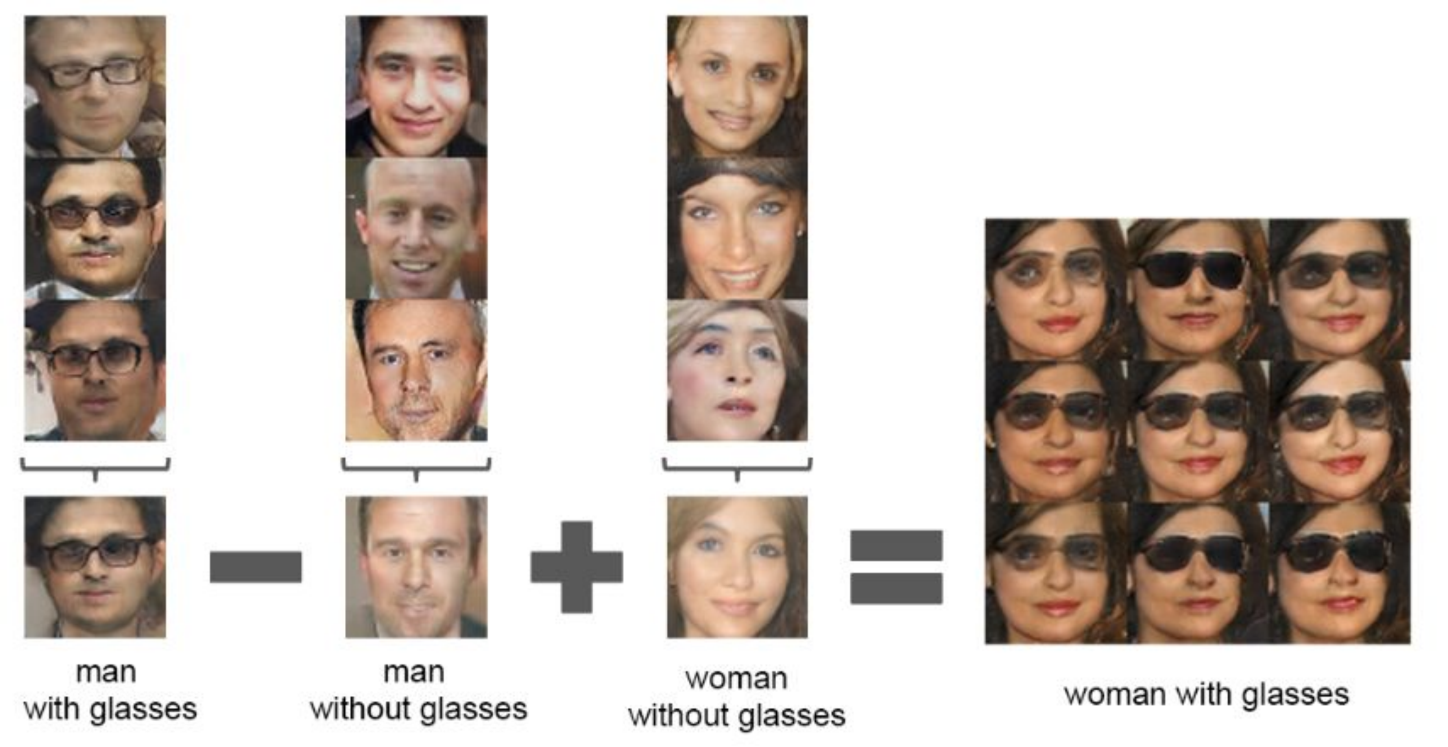
Face Generation

Figure 10: More face generations from our Face DCGAN.

Interesting Vector Arithmetic Representations



One more example...



Thank You