GAN for cat faces

Aleksandra Leszczyk

Abstract
We generate images of cat faces using deep convolutional generative adversarial network (GAN).

Keywords
GAN — machine learning

Contents

1 Objective 1
2 Description of data set 1
3 Data exploration, cleaning, and feature engineering 1
4 Model description 2
4.1 Discriminator 2
4.2 Generator 2
4.3 Unsupervised learning and optimization 2
5 Results for different models 3
5.1 Loss function values 3
5.2 Latent size = 64, slope = 0.2 3
5.3 Latent size = 128, slope = 0.2 4
5.4 Latent size = 256, slope = 0.2 4
5.5 Latent size = 128, slope = 0.1 5
5.6 Model evaluation 5
5.7 Model recommendation 5
6 Summary and key findings 5
7 Future plans 5

1. Objective

Since video games became part of life and people spend more and more time in virtual reality. Graphic designers create wonderful fake worlds that are the feasts for our eyes. Such the work consumes time and resources, so, 3D models and textures are reused. It makes the virtual worlds less attractive since the shapes and patterns in real world are seldom exactly the same. Creating a generator that can produce realistic and different images can enhance the user experience in virtual world. In this work, we make a step towards this goal. Our objective is creating a model that generates realistic cat faces.

2. Description of data set

We use data from Cats faces 64x64 (For generative models) repository. The data set contains 15747 color images of cat faces. The size of a single cat picture is 64×64 pixels, so, one observation contains 12288 features. The following figure presents an example subset of cat face images.

3. Data exploration, cleaning, and feature engineering

Since data contains images, data exploration, e.g. examination of descriptive statistics and visualization of data distributions does not provide any insight. Data cleaning is impossible as we do not have outliers or missing values. Feature engineering is not required for this model because we expect that a neural network identifies non-linear dependencies. If the learning process is slow, it can be beneficial to add noise to real and generated data to enhance the learning process of the discriminator.

4. Model description

We use generative adversarial network framework in which we define two deep convolutional neural networks, called generator and discriminator. We examined four neural network models with different parameters. The general structure of discriminator and generator are described in detail in Subsection 4.1 and 4.2. First, we test three models where a generator has different size of a latent layer – 64, 128, and 256 elements. For these models, we use a leaky ReLU activation function with slope parameter equal to 0.2 in the discriminator network. Then, we check the effect of changing the discriminator's slope parameter by setting it to 0.1 in a model with latent size equal to 128.

4.1 Discriminator

The discriminator is a neural network that tries to recognize fake and real images. Thus, its input is a 12288-element vector that corresponds to a color 64×64 pixel image. The layers and activation functions of a discriminator are listed as follows:

1. 2D convolutional layer with 4×4 kernel with 3 input channels and 64 output channels
2. Batch normalization layer
3. Leaky ReLU activation function with 0.2 (or 0.1) negative slope
4. 2D convolutional layer with 4×4 kernel with 64 input channels and 128 output channels
5. Batch normalization layer
6. Leaky ReLU activation function with 0.2 (or 0.1) negative slope
7. 2D convolutional layer with 4×4 kernel with 128 input channels and 256 output channels
8. Batch normalization layer
9. Leaky ReLU activation function with 0.2 (or 0.1) negative slope
10. 2D convolutional layer with 4×4 kernel with 256 input channels and 512 output channels
11. Batch normalization layer
12. Leaky ReLU activation function with 0.2 (or 0.1) negative slope
13. 2D convolutional layer with 4×4 kernel with 512 input channels and 1 output channel
14. Sigmoid activation function

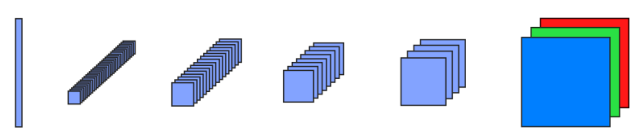
Here, we use batch normalization that introduces regularization and speeds up learning process. As the alternative, dropout method could be used.

input → 1, 2, 3 → 4, 5, 6 → 7, 8, 9 → 10, 11, 12 → 13, 14

4.2 Generator

The generator is a neural network that tries to generate cat images that look realistic for discriminator. The input is a vector with 64, 128, or 256 elements.

1. 2D convolutional layer with 4×4 kernel with 64 / 128 / 256 input channels and 512 output channels
2. Batch normalization layer
3. ReLU activation function
4. 2D convolutional layer with 4×4 kernel with 512 input channels and 256 output channels
5. Batch normalization layer
6. ReLU activation function
7. 2D convolutional layer with 4×4 kernel with 256 input channels and 128 output channels
8. Batch normalization layer
9. ReLU activation function
10. 2D convolutional layer with 4×4 kernel with 128 input channels and 64 output channels
11. Batch normalization layer
12. ReLU activation function
13. 2D convolutional layer with 4×4 kernel with 64 input channels and 3 output channels
14. Hyperbolic tangent activation function



input → 1, 2, 3 → 4, 5, 6 → 7, 8, 9 → 10, 11, 12 → 13, 14

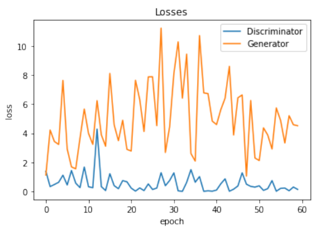
4.3 Unsupervised learning and optimization

In the optimization process, we use binary cross entropy loss function. To minimize the loss function, we use Adam optimizer with learning rate set to 0.002 and average decay rates of average of gradients set to 0.5 and 0.999. We train neural networks for 60 epochs.

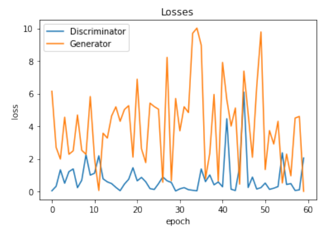
5. Results for different models

5.1 Loss function values

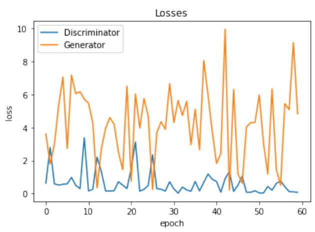
The following figure shows discriminator and generator loss function values along the optimization process for different neural network parameters – latent layer size in generator and slope parameter in leaky ReLU activation function in discriminator neural network.



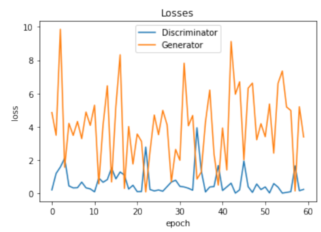
latent size = 64, slope = 0.2



latent size = 128, slope = 0.2



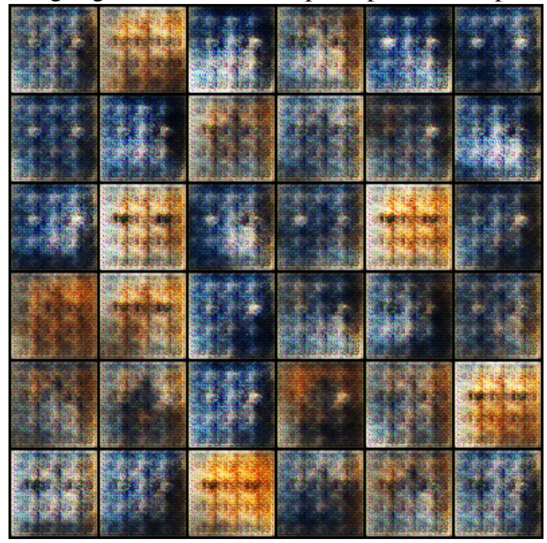
latent size = 256, slope = 0.2



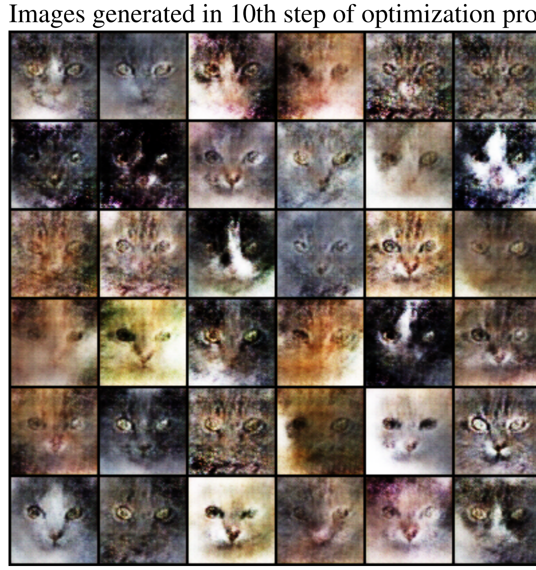
latent size = 128, slope = 0.1

5.2 Latent size = 64, slope = 0.2

Images generated in 2nd step of optimization process



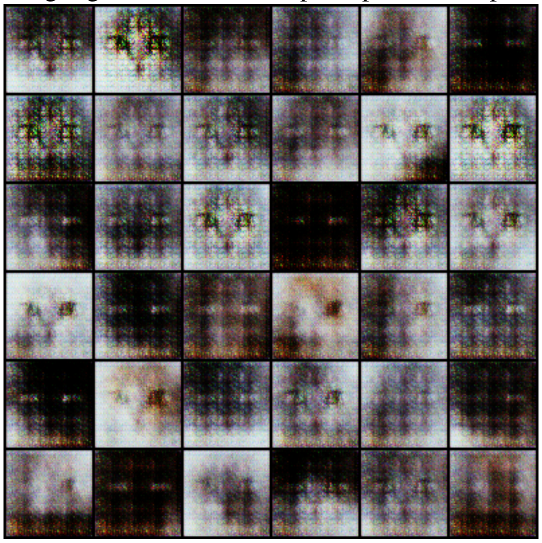
Images generated in 10th step of optimization process



Images generated in 60th step of optimization process

5.3 Latent size = 128, slope = 0.2

Images generated in 2nd step of optimization process

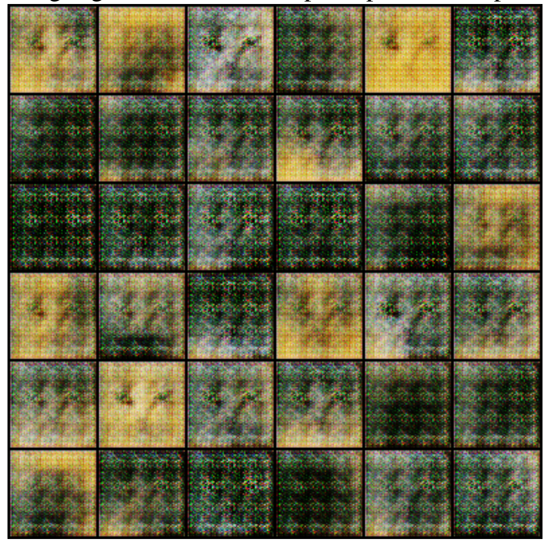


Images generated in 10th step of optimization process

Images generated in 60th step of optimization process

5.4 Latent size = 256, slope = 0.2

Images generated in 2nd step of optimization process

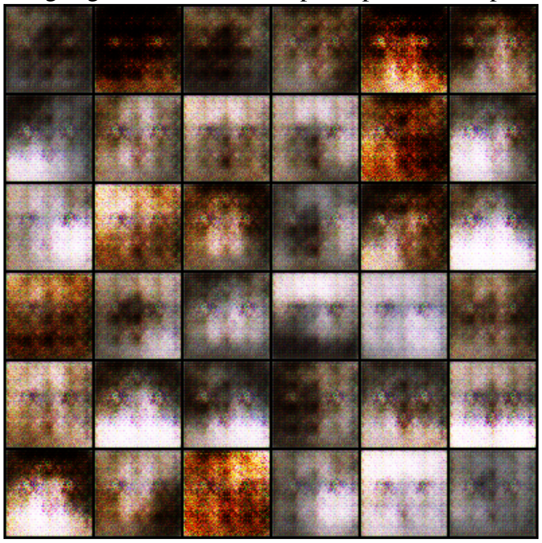


Images generated in 10th step of optimization process

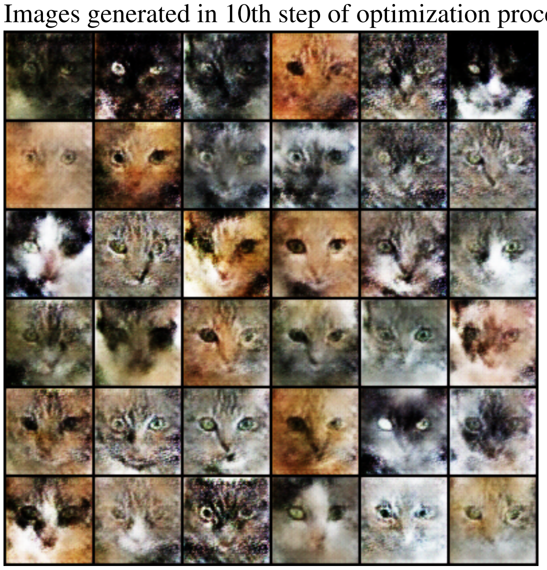
Images generated in 60th step of optimization process

5.5 Latent size = 128, slope = 0.1

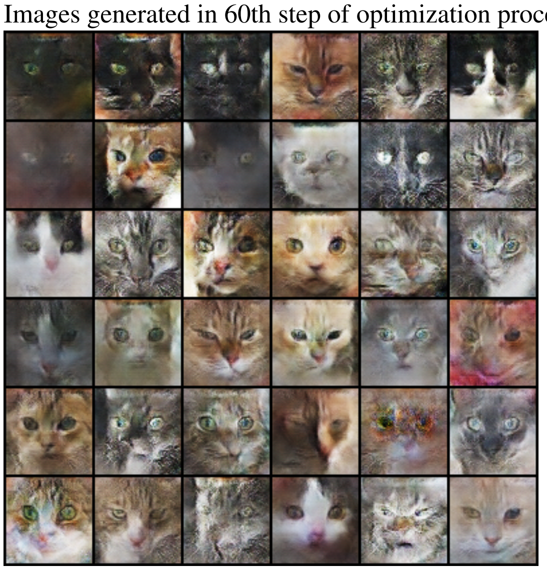
Images generated in 2nd step of optimization process



Images generated in 10th step of optimization process



Images generated in 60th step of optimization process



5.6 Model evaluation

Since the discriminator learns to recognize real and fake images during the learning process together with improving the generator, the loss values do not provide a good estimate of model performance. Thus, we use a human labeling method. We asked an independent human tester to evaluate a number of real cat faces in each set of fake images. The human tester has got recently a training in recognizing cats and is 4 years old. We obtained the following results:

latent size	slope	number of realistic fake images
64	0.2	1
128	0.2	4
256	0.2	10
128	0.1	2

5.7 Model recommendation

We observe that model with a latent size equal to 256 and with slope = 0.2 generates the most realistic cat faces. Smaller latent size and lower value of a slope in leaky ReLU activation function decreased the quality of the output. Thus, among the tested models, we recommend a model with latent size = 128 and leaky ReLU slope = 0.2 for generating images.

6. Summary and key findings

The main objective of this project is a generation of realistic cat face images. We obtained fake pictures that included typical cat features - almond-shaped eyes, nose, mouth, fur patterns, and color. However, most of them have some type of deformation (e.g. irregular eyes), unclear edges, or noise.

We tested four variations of deep convolutional GANs. We checked models that were differing with the size of the latent layer of the generator, and a slope parameter of the leaky ReLU activation function of the discriminator. We observed that increasing a latent size increased the number of realistic generated images as evaluated with the human labeling method. Increasing a latent size might introduce over-fitting and problems with generalization, but we do not observe this effect for tested values. Thus, the optimal size can be larger than 256.

7. Future plans

Although generated cat faces have cat features, our human labeling methods indicate that most of them do not look realistic even in our best model. Thus, action is required. We can improve a generator by increasing a size of a latent layer. A solution that requires more research is to find an optimal size of a kernel or add a hidden layer. Also, we can enhance the discriminator's capability to recognize fake figures by adding a hidden layer or changing kernel size. Training can be improved by using other cost functions or by adding noise to real images. We have many possible solutions, but each of them requires testing or adjustment.