WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF MATHEMATICS AND INFORMATION SCIENCE

Project III Report

Deep Learning

Students:
Karina Tiurina (335943)
Mateusz Zacharecki (313549)

Supervisor:
mgr inz. Maciej Żelaszczyk

Warsaw 2024

Contents

1 Research problem

The topic of the project is Image generation with diffusion models based on 10% of the LSUN bedroom scene dataset:
https://www.kaggle.com/datasets/jhoward/lsun_bedroom
The dataset contains 303125 images of bedroom scenes. This data is particularly useful for generative models.

The aim of the project is to test and compare different network architectures, including architectures that converge to generate satisfactory images. Moreover:

- calculate the Fr´echet Inception Distance (FID) for your generated images and compare it to results from literature
- assess your results qualitatively
- investigate the influence of hyperparameters on obtained results
- discuss sets of hyperparameters which help in overcoming training collapse and mode collapse
- select two of your generated images together with their latent noise matrix; interpolate linearly between the two latent matrices to generate 8 additional latent matrices; use these 8 matrices to run the diffusion process and generate images from your model; present the 10 generated images (8 newly generated and 2 generated previously) and discuss the importance of the results
- discuss any additional findings.

2 Application instruction

Source code has the following structure.

- diffusion
    - train.ipynb
- gan
    - train.ipynb
- vanilla_pixel_diffusion
    - train.ipynb

To reproduce the results presented in this report, please, run /diffusion/train.ipynb, /gan/train.ipynb and /vanilla_pixel_diffusion/train.ipynb.

3 Theoretical introduction

Diffusion models are advanced machine learning algorithms that uniquely generate high-quality data by progressively adding noise to a dataset and then learning to reverse this process. This innovative approach enables them to create remarkably accurate and detailed outputs, from lifelike images to coherent text sequences. Central to their function is the concept of gradually degrading data quality, only to reconstruct it to its original form or transform it into something new. This technique enhances the fidelity of generated data and offers new possibilities in areas like medical imaging, autonomous vehicles, and personalized AI assistants.

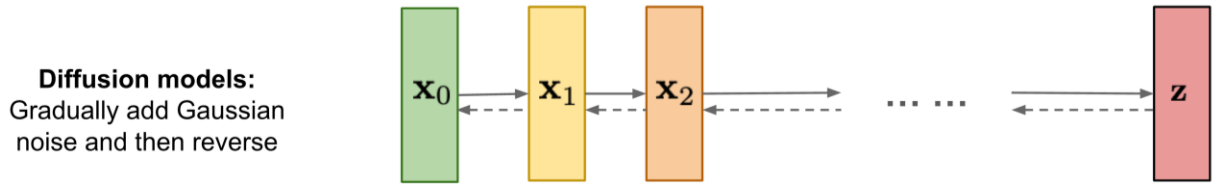

Figure 3.1 Diffusion architecture

Generative Adversarial Networks, or GANs, are a deep-learning-based generative model. More generally, GANs are a model architecture for training a generative model, and it is most common to use deep learning models in this architecture. The GAN architecture was first described in the 2014 paper by Ian Goodfellow, et al. titled "Generative Adversarial Networks." 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.

- Generator - Model that is used to generate new plausible examples from the problem domain.
- Discriminator - Model that is used to classify examples as real (from the domain) or fake (generated).

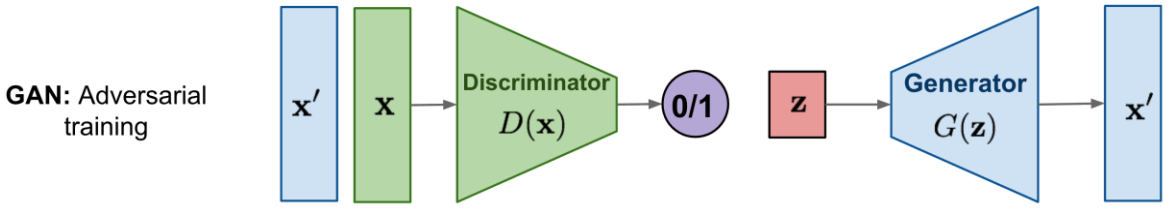

Figure 3.2 GAN architecture

A vanilla pixel diffusion model is a type of generative model used in machine learning, particularly in the realm of image generation. It leverages the diffusion process to generate images by iteratively refining a noisy image towards a clean image.

Core Concepts:

1. Diffusion Process: Adds noise to an image in a forward process and then reverses this process to generate images.
2. Noise Schedule: Controls how noise is added and removed at each step.
3. Denoising Network: A neural network trained to remove noise step by step.

Steps:

1. Initialization: Start with a noisy image.
2. Forward Diffusion: Add noise to an image over several steps.
3. Training: Train the network to predict and remove noise.
4. Reverse Diffusion: Iteratively denoise the image to generate a clean image.

4 Conducted experiments

4.1 Diffusion model

Unfortunately, none of the tested diffusion models were able to provide a ‘bedroom’ looking like image.

The loss function is able to drop significantly during the first epoch.

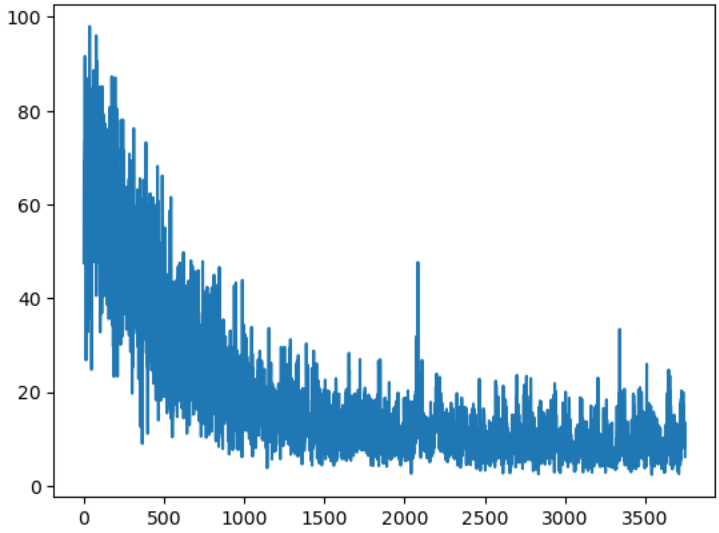

Figure 4.1 Diffusion model loss function

However, all predicted images were either too blurry or do not contain anything reasonable. We applied the following updates to the model:

1. Trained different image sizes: (256x256, 128x128, 64x64)
2. Colorful images and grayscale images
3. Batch size 16, 8, 4.

We would like to also test augmentation, but unfortunately, all our experiments on that lead to “OutOfMemory” exception.
Figures below contain examples of fake images predicted by diffusion models.

Figure 4.2 Diffusion model trained on 150k images; 10 epochs; batch size 8

Figure 4.3 Diffusion model trained on grayscale images; batch size 4

4.2 GAN

Tests on the Generative Adversarial networks were more successful, so we decided to run additional experiments on them.

We tested the following changes to the GAN model:

1. Batch sizes: 128, 64, 32, 16;
2. Activation functions on both Genarator and Discriminator models: ReLU, LeakyReLU, ReLU6, ELU.

Figures below show comparison of fake images generated by models trained on different batch sizes and activation functions.

Figure 4.4 GAN; batch size 64

Figure 4.5 GAN; batch size 128; ReLU6

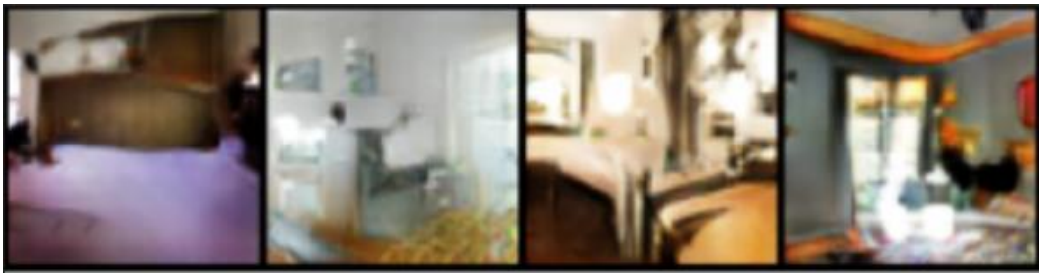

Figure 4.6 GAN; batch size 32

Figure 4.7 GAN; batch size 16

Figure 4.8 GAN; batch size 128; ELU

FID score calculated during GAN training on batch size 128 is shown below.

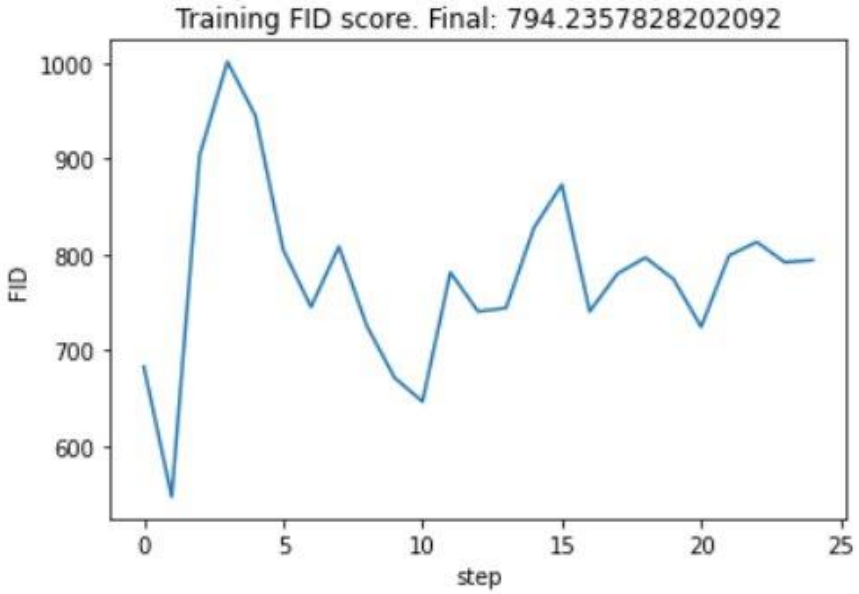

Figure 4.9 FID score of GAN

5 Summary

Required computing power for efficiency of training process and limited access to GPU led to several issues in gaining reasonable or even any (like in vanilla pixel diffusion) results.

The following research could be further conducted to improve learning accuracy.

1. Test more hyperparameters, both related to training process and related to regularization.
2. Test other architectures, e.g. DDPM - Denoising Diffusion Probabilistic Models, Improved DDPM, Stable Diffusion, StyleGAN.
3. Increase number of epochs in training process.

References

[1] https://www.kaggle.com/datasets/jhoward/lsun_bedroom

[2] https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

[3] https://www.superannotate.com/blog/diffusion-models

[4] https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

[5] https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html