Assignment #5 – Generative Models

https://github.com/Lemorita95/1FA006/tree/main/5_generative-models

Written summary

Also available in https://github.com/Lemorita95/1FA006/blob/main/5_generative-models/README.md

approach

1. hyperparameters:

   batch_size = 64

   learning_rate = 2e-4

   num_epochs = 50

   latent_dimension = 128

   optimizer momentum β_1 = 0.5

2. Define a transform that converets image to tensor and normalize it with mean=0.5 and std=0.5 (convertes image range [0, 1] to [-1, 1], suitable for tanh activation function of generator);
3. Load data with torchvision dataset and apply transform;
4. Load dataset into a DataLoader object;
5. To develop the models, their architecture were based on the article (Radford & Metz, 2016) https://arxiv.org/pdf/1511.06434;
6. Instantiate Discriminator() with number of channels of image (1 for MNIST, greyscale);
7. Instantiate Generator() with latent_dimension and number of channels of image;
8. Define optimizer with same learning rate and $ \beta_1 $ for Discriminator and Generator;
9. Define loss function (BCE for binary classification);
10. Train model and log training progress in tensorboard;

results

1. the training loop for the hyperparameters took around 2.5 hours;
2. the training had convergence;

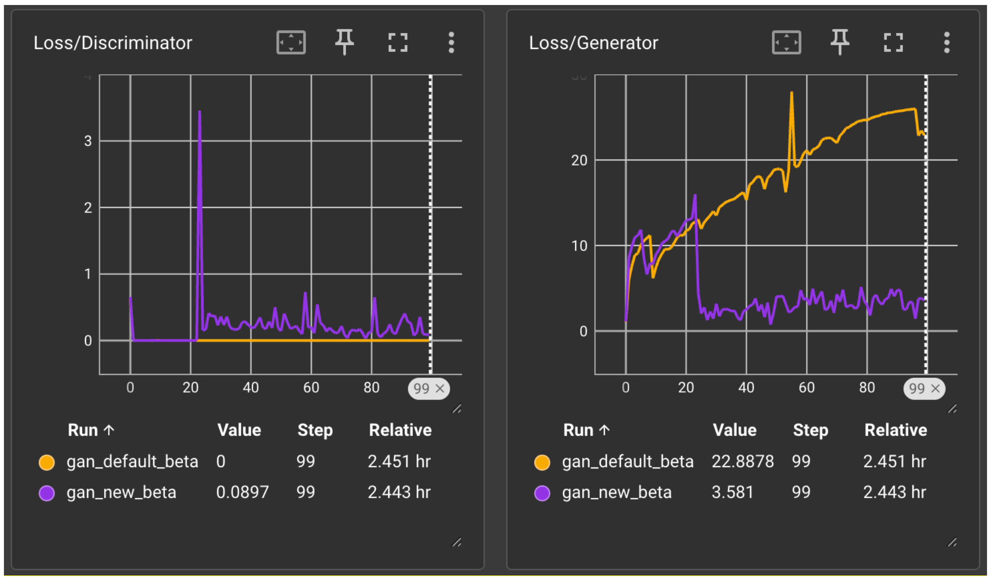
3. the generator is able to generate images of digits, altough some digits might look clearer for an human interpretation (numbers 0, 3, 6, 7, 9);

challenges

1. Computational expensive to train the network (almost 3h);
2. Runned the model with default values of Adam optimizer $ \beta_1 $ and the train did not converge, the discriminator dominated at the adversarial approach (very low loss but high loss for generator);
3. Then, following the guidelines of (Radford & Metz, 2016), page 3, "Details of Adversarial Training", the training process did converged when changing the $ \beta_1 $ parameter to 0.5 at Discriminator and Generator optimizers;
4. Further improvements that could be made (not tested):
    - decrease the learning rate of Discriminator (since its dominating the training).
    - change the models architecture (more complex for Generator and less complex for Discriminator).
    - implement Wasserstein distance as the loss function and remove the final sigmoid activation from the model.

Result plots

The main objective of this section is to present the results of adversarial training with emphasis on the optimizer momentum β_1.



Run ↑	Value	Step	Relative
● gan_default_beta	0	99	2.451 hr
● gan_new_beta	0.0897	99	2.443 hr

Run ↑	Value	Step	Relative
● gan_default_beta	22.8878	99	2.451 hr
● gan_new_beta	3.581	99	2.443 hr

Figure 1: Train and validation loss for default optimizer $\beta_1 = 0.9$ *and new value* $\beta_1 = 0.5$.

From Figure 1 we can see that using the default value of momentum for Adam optimizer leads to instability in the training, reducing the value helped stabilize the training (Radford & Metz, 2016). For the new scenario, in the beginning of the training, the discriminator become too good to fast in comparation to the generator, as seen by the increasing generator loss, at the 24th step the dynamic changed, and the training stabilize (or at least did not got unstable).

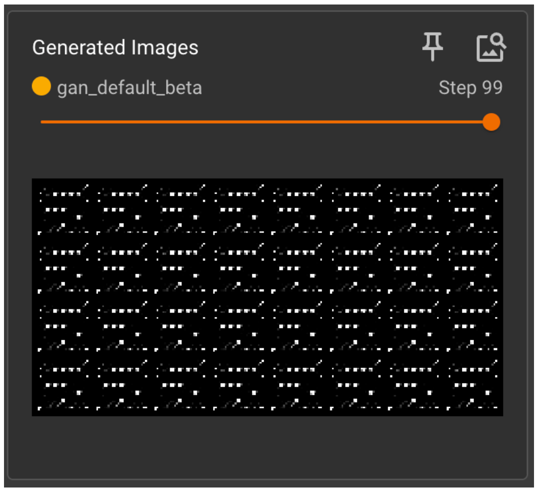


Figure 2: Generated images after finishing training for $\beta_1 = 0.9$

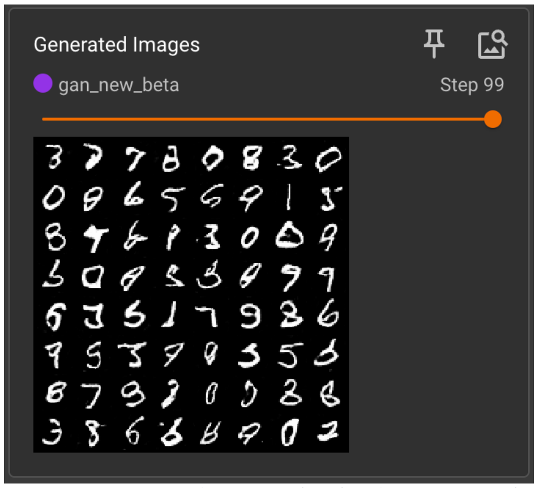


Figure 3: Generated images after finishing training for $\beta_1 = 0.5$

From Figure 2, show that the generator did not produce good results from the first training ($\beta_1 = 0.9$) as seen by the mode collapse. Figure 3 shows better results although some digits might appear clearer for human recognition than others.

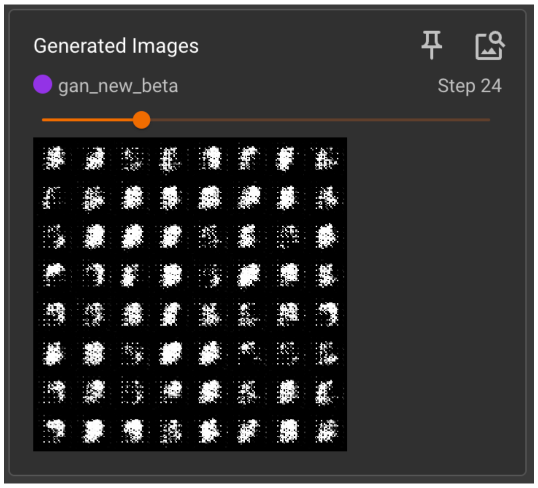


Figure 4: Generated images after finishing training for $\beta_1 = 0.5$ at the 24th step.

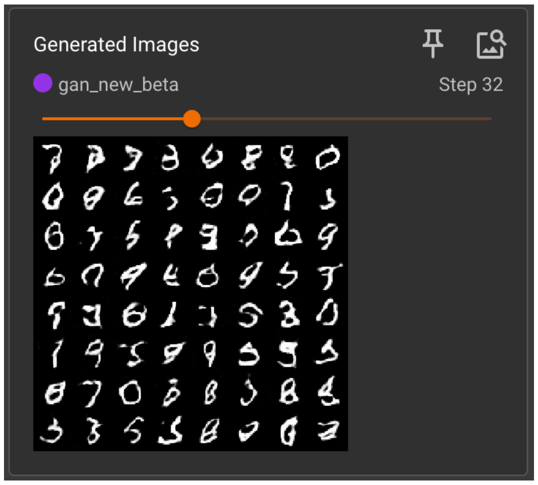


Figure 5: Generated images after finishing training for $\beta_1 = 0.5$ at the 32nd step.

Figure 4 and Figure 5 show how the output images quality increased (from human perspective) after the 24th step, it shows the effect of the decrease in the generator's loss, observed in Figure 1.ß

References

Radford, A., & Metz, L. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.