Question 2: Custom GAN Image Generation *

*Using cats and dogs images from CIFAR dataset

Areeba Fatah
AI (artificial intelligence)
FAST, NUCES)
Islamabad, Pakistan
i210349@nu.edu.pk

I. Introduction

This report describes the implementation of question 2 i.e. custom Generative Adversarial Network (GAN) using the CIFAR-10 dataset with only cats and dogs classes.The discriminator architecture is inspired by Siamese network. The custom discriminator is designed to compare a real and a generated image ,returning a similarity score which is different than classifying them as real or fake. The generator is the same i.e. it has to minimize the similiarity score to generate close enough images.

II. Methodology

A. Dataset

The dataset was the CIFAR-10 , which contains 60000 32x32 RGB images across 10 classes but here only cats and dogs classes were used. w

- **Number of images per class**: 6000
- **Classes**: Cats and Dogs.
- Total images=12000

B. Preprocessing

The following is the detail of preprocessing steps followed. i have used tesndorflow since it was faster than pytorch.

- **Filtering of classes:** with simple python.
- **Normalizing Images**: The pixel values of images were normalized to a range of tanh to stabilize training.
- **Image Resizing:** Images were resized to 32x32.
- **Shuffling and slicing:** Data was sliced into batches of size =64
- **Data Preparation function:** All the above was arranged in a function to provide organized pipeline.

C. Model Architecture

D. Generator

The generator was the traditional one having:

- Fully connected layer followed by batch normalization and ReLU activation.
- Several ConvTranspose2D to scale up the input from a noise vector to a 32x32 image (input image).
- Tanh activation in the final layer to output pixel values in range of (minus 1 and 1).

E. Discriminator

The discriminator is different from a traditional one since it was inspired from a Siamese network architecture that takes two inputs and return a similarity score between them. The discriminator rather then classifying the image as real or fake also returns a similarity score. It's goal is still the same though. The discriminator consists of:

- 2 parallel convolutional branches,one for each image with Conv2D layers, followed by ReLU activation and Batch Normalization.
- The outputs of both branches are concatenated and passed through several fully connected layers.
- A final layer outputs a similarity score. The generator aims to minimize this score and the discriminator tries to maximize it.

1) Key Features: :

- **Input**: A pair of images.
- **Output**: A similarity score between the two images.

F. Training

1) Loss Function: :

- **Generator Loss**: It is designed to minimize the similarity score.
- **Discriminator Loss**: Its goal is to maximize the dissimilarity between the real and generated images.
- **Optimizer**: Adam optimizer with a learning rate of 0.0001 is used.
- **Epochs**: 200 epochs.
- **Batch Size**: 64.

III. Results

1) Outputs: Following are some of the outputs:

Fig. 1. Epoch 701

Fig. 2. Epoch 1001

Fig. 3. Epoch 901

*2) **Training Loss**:* The model was trained iteratively with 200 epochs at a time with alternate weights updation.

- **Generator Loss**: The generator's loss was at 0.711 in the last epoch.
- **Discriminator Loss**: The discriminator's loss was at 1.4659.
- Both of their curves were not smooth as convergence issues occurs.

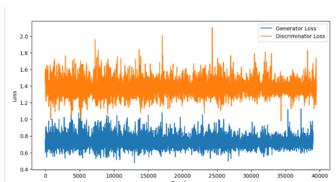

Fig. 4. Losses

IV. Discussion

The main challenge was to stabilize the GAN's training which is very hard as it has convergence issues 4 as shown here the curve is not smooth. The use of a Siamese-like network architecture in discriminator increased its complexity, but it gave better feedback for the generator to improve over time. The overall performance was good. The coherence of the generated images improved gradually as the generator learned to minimize the similarity score but very clear images were not given. Tuning of the learning rate helped improve training stability.

Conclusion

This custom GAN following siamese like network was able to generate somewhat realistic images of cats and dogs. However more fine-tuning is needed to generate higher quality results and maybe not normalize the images to avoid bluriness.

Prompts

- Change this GAN to have siamese network architecture in discriminator.
- Alternate the wieghts updation.
- Write report for this code.
- Why does GAN not converge smoothly.
- Switch this code to tensorflow.
- give me preprocessing pipeline.
- Follow the steps given in the assignment.
- Design the training loop and display the results in training along with losses.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672-2680, 2014.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[3] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto, Tech Report*, 2009.