Generative Adversarial Networks

Project 3

Deep Learning in Computer Vision

June 2021

The main goal of this project is to implement a CycleGAN that can convert images of horses to zebras, and vice versa.

MNIST

You have also been working on creating networks that can generate digits from the MNIST dataset. If you are in need of more content for your poster, you are allowed to also include these on your poster. For this you could: show results from these networks, describe their architectures, the training process, and describe your process of making these choices.

Data

We have supplied you with a dataset for this task, located on the following link.
https://courses.compute.dtu.dk/02514/project_data/horse2zebra.zip
The dataset contains two folders: train and test. Inside each of these folders there are two folders: A and B, where A contains pictures of horses, while B contains pictures of zebras.

Hints

- The datasets don't have the same number of images, and you can handle this however you please.
- Some of the images are not in color, and will need to be converted to have three channels first. You can do this with `Image.open(image_path).convert('RGB')`
- Save your model and optimizer state to your Google Drive often, so you won't lose progress if the runtime disconnects.
- Save examples of generated images to Google Drive after every epoch.
- You shouldn't use `.backward(retain_graph=true)`. If you run into needing to use it, you should look at which times you're calling .backward() on which tensors, and understand why it's trying to backprop through your model multiple times.
- Normalize your images to between -1 and 1, as the generator uses a `tanh()` as its final activation.
- Start out with using 128×128 images, as its faster to train your model and find possible mistakes this way.
- Remember to weigh your loss functions with $\lambda_{identity}$ and λ_{cycle}

CycleGAN

Implement and train a CycleGAN to convert horses to zebras and vice versa, evaluate its performance, and document the process. For details of CycleGAN and a suggested network architecture, see `Lecture 3.2 CycleGAN.pdf`, and the CycleGAN paper.

Some questions you could answer on your poster are: (these are suggestions and not mandatory)

- How is your model performing? Is it able to carry out the desired task? Only on some images? Do you notice common things in the images it does good/bad?
- How is the performance of the discriminator networks compared to the generator? How big are the contributions from each loss term? Does this change over the epochs?
- Try training your discriminator with pictures generated by previous versions of the generator, in addition to the current ones.

  > Keep track of the last 50 generated images, and randomly select one of these to also compare against when updating the discriminator. Similar to section 2.3 in this paper

- How does a different GAN loss affect the training?
- Increase the number of residual blocks in the generator. 9 is a good choice.
- If you feel confident in your setup, try training it on 256×256 images.
- Training a CycleGAN on different domain translation problem?
- What are the strengths and weaknesses of the cyclic loss? Think of what happens if all stripes are removed from a zebra.
- Compute the Frechet Inception Distance (FID) of your generated images.

Hand-in

Your process, performance evaluation and results should be documented and discussed in a PDF poster to be uploaded on DTU Learn. The deadline for this is Tuesday 23:59 (midnight).

Additional task

You should give peer feedback on 3 other posters after handing in. This should be done at latest Wednesday at 11:59 (noon). You find these assigned to you in Peergrade.