Home Assignment 02

Deep Learning Foundations — Due by **8th January, 2025 at 23:59 CET**

Sandipan Sikdar, Jonas Wallat, Tobias Kalmbach — Winter Semester 2024/25

Submission Details

- Please submit a single compressed folder (.zip) containing one PDF file with all text responses and one Jupyter Notebook per coding task. Please keep all generated outputs in the Jupyter Notebooks so we can grade your submission without rerunning your code.
- For each answer/code snippet, clearly state what task you are referencing. If you are not confident in an answer, try nonetheless, as it is possible to receive partial points.
- You may work in groups of up to three students. Add all team members' names and matriculation numbers at the top of the PDF file **and** your notebooks.
- You may use online resources for help, but copying online resources is prohibited. Cite or link any resources you use.
- If you have any questions, create a post in the forum.

1 DCGAN [45 points]

DCGANs have been introduced in the lecture. Explain the DCGAN architecture and their role in image synthesis (1 paragraph).

1.1 Data Loading and Preprocessing

- The dataset for this assignment is Fashion-MNIST, a collection of Zalando's article images[1]. The dataset can be accessed via the torchvision.datasets module. In case of problems with training time and computation, it is fine to work on a subset of Fashion-MNIST.
- Implement a data loading pipeline for DCGAN training. Preprocess the data (normalize) and prepare it for DCGAN.
- Visualize some samples of different classes of the dataset to get familiar with the data, format, and dimensions.

1.2 DCGAN Architecture

- Design the DCGAN architecture for image synthesis to generate new images similar to the Fashion-MNIST dataset.
- Implement the generator and discriminator architectures of the DCGAN architecture.
- Discuss the rationale behind your design choices.

[1] https://github.com/zalandoresearch/fashion-mnist

1.3 Training

- Try to find the best initialization of the weight parameters and check if it affects the training procedure.
- Implement the training loop for DCGAN and train the model.
- Explain the adversarial loss used to train a GAN.
- Include a mechanism for monitoring the training progress using Tensorboard[2]. (1) Plot the losses of generator and discriminator versus training iterations. (2) Visualize the generator's output on a batch of fixed noise for each epoch.
- Take screenshots of the outputs and plots generated by Tensorboard and upload them with your submission.

1.4 Evaluation

- Qualitatively evaluate the generative performance of your DCGAN by visualizing some of the newly generated samples.
- Address any challenges you encountered during training.

2 VAE [40 points]

In this task, you will implement a Variational Auto Encoder (VAE), as presented in the lecture and exercise. You will train the VAE on the CIFAR10 dataset, comprising 32x32 color images in 10 classes. The dataset is available via the torchvision.datasets module[3]. You are free to use any framework or library of your choice, but you will need to do the following:

1. Load the dataset and display some examples. Use a subset, if you run into memory issues.
2. Implement an encoder and decoder and combine them to form a VAE (see also Exercise Week 6).
    - As the images are a different size, you cannot copy the architecture directly. Also note that CIFAR10 has three color channels, while MNIST only has one!
    - You may use any number (> 1) of latent dimensions.
    - Briefly explain your design choices, including the chosen number of latent dimensions.
3. Explain the loss function you use.
4. Train the VAE for a few epochs and compare some input and reconstruction images after each epoch.
    - Note that the reconstructed images may not be as high of a quality as the input images. You should be able to make out general structures, but not specific details. The focus of the task is to implement a VAE yourself on a new dataset, not the reconstruction quality. See Figure 1 for a reference of what reconstruction quality is sufficient.

[2] https://www.tensorflow.org/tensorboard
[3] https://pytorch.org/vision/master/generated/torchvision.datasets.CIFAR10.html

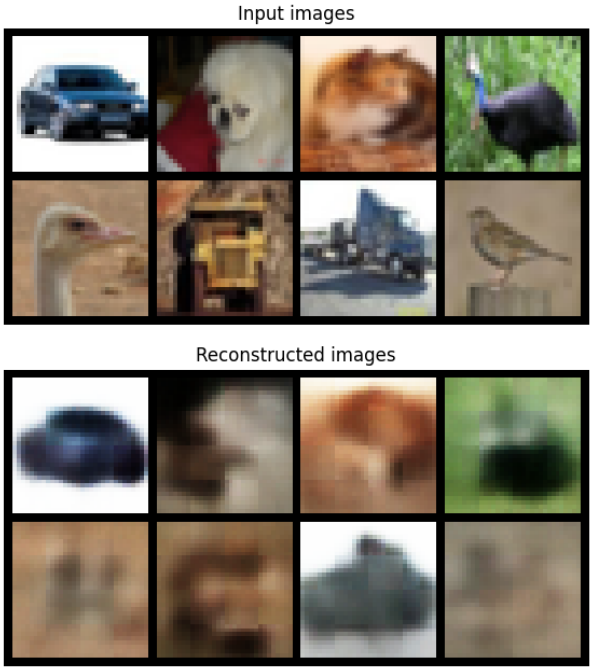

Figure 1: VAE reconstruction results

3 Pen & Paper Exercises [15 points]

1. Reparametrization trick: Explain how the trick works and what problem it addresses.
2. Explain in your own words why optimization is hard in the case of GANs. What are common ways to improve GAN training stability?
3. Shortly compare GANs and VAEs. In which setting would you select one over the other?