

Marked exercises, lecture 10

This is the set of marked exercises for Submission 2. In this assignment, you will get familiar with Wasserstein Generative Adversarial Network (WGAN) and with Denoising Diffusion Probabilistic Model (DDPM). Submission 2 consists of completing the implementation of a WGAN training process, training the WGAN, completing forward and reverse diffusion processes, training and sampling for DDPM and answering written questions. In total you can get 60 points for Submission 2.

Note:

- You are **not** allowed to change the code that we provide. All of the definitions of functions and all the hyperparameters should be kept as they are. You are **not** allowed to create new functions and you are **only** allowed to write your code wherever you see `raise NotImplementedError`.
- Keep your written answers within the specified word limit. Exceeding the limit will result in a penalty.

WGAN [17 points]

For this part of the submission, please refer to the files in `WGAN/` folder.

2.10.1 [2 points] Implement weight clamping in `trainer.py`: `def clamp_weights()`.

2.10.2 [5 points] Implement Discriminator update step in `trainer.py`: `def disc_step()`.

2.10.3 [5 points] Implement Generator update step in `trainer.py`: `def gen_step()`.

2.10.4 [5 points] Complete the call of `disc_step()` and `gen_step()` in `train_epoch()` in `trainer.py`. Train your WGAN for 8 epochs. Report the generated images.

DDPM [29 points]

For this part of the submission, please refer to the files in `DDPM/` folder.

2.10.5 [5 points] Implement the forward diffusion process: `forward_diffusion_sample()` and a cycle for noising the image with `generate_noisy_image()`.

2.10.6 [5 points] Implement training.

2.10.7 [15 points] Implement the reverse diffusion process:

- Posterior P mean and variance, where the noise is predicted from the model: `p_posterior_mean_variance()` and variables in `__init__`.
- Generate images: `generate_images_at_previous_timestamp()` and generation loop with `generate_images()`.
- Posterior Q mean and variance: `q_posterior_mean_variance()`, dependent on `x_0` and `x_t`.

2.10.8 [4 points] Train the model and sample images.

2.10.9 [6 points] Analyse the impact of skipping time steps during the generation process (generation of images with reverse diffusion process). Provide your hypothesis of how this might influence the final output. Then, complete the `generate_images_along_timestamps()` function, and test what happens in your case (with your trained model on the Fashion MNIST dataset), write if the results met your expectations and why. Provide your answer in `S2_<YOUR_SCIPER>.pdf` file (150 words max).

EE-559 Deep Learning

Set-ups:

- skipping steps 5-75
- skipping steps 130-200

2.10.10 [8 points] The training of Denoising Diffusion Probabilistic Models often replaces the loss L_{VLB} with a simplification of the loss, L_t^{simple} DDPM (see [the original](#) paper). Explain how this simplified objective relates to L_{VLB} and why it is considered a valid alternative.

For this exercise, you are asked to use ChatGPT as a conversational assistant to help you understand the underlying theory. You need to include your conversation history with ChatGPT as a part of your submission, demonstrating how your understanding developed throughout the discussion. As a result of your conversation, provide your own explanation in S2_<YOUR_SCIPER>.pdf file (250 words max) on how the simplified objective relates to L_{VLB} and why it is considered a valid alternative.

When you make a statement, you need to ensure that it is supported with an academic resource. Even though you use ChatGPT for understanding the theory, you need to support the claims with **peer-reviewed academic** resources. Standard referencing practices apply. Use IEEE referencing style. Not referencing a used source will be penalised.

See an example of how to share the conversation below.

“Share ChatGPT Conversation”: <https://chatgpt.com/share/681108a6-4d78-8006-87a9-a94d16df6a92>