Assignment 4 - Multi Layer Perceptron

Advanced Topics in Neural Networks

24 October 2023

Homework

Deadline End-of-Day 30 October 2023

Develop a training pipeline with the following components:

1. Implement a custom PyTorch Dataset class to read the homework dataset found in the Lab04 folder ("Homework Dataset"). Investigate the folder/file structure and naming convention used. (1 point)

   It needs to follow the following specifications:

   - It should receive a folder path of where the dataset is stored.
   - When iterating the Dataset, it should return tuples of the form: $(start\_image, end\_image, time\_skip)$
   - The *start_image* is an image from location L taken at time T_1.
   - The *end_image* is an image from the same location L taken at time T_2.
   - The *time_skip* is a non-zero, positive integer value, that signifies the number of months between T_1 and T_2.
   - The Dataset needs to accept an optional parameter for transformations to be applied to the images.
   - You need to define and compute the dataset size.
   - **Bonus (1 point)**: Use a random rotation augmentation for your dataset. Be careful to make sure you apply the same rotation to both images!

   Practically, two images from the same location, taken at *time_skip* months apart.

2. Define and test three PyTorch DataLoaders, one for training, one for validation, and one for testing. The data isn't split, you can choose how to define the data splits, restructure the folders or use additional csv files. As a general indication, an okay split is: 70% train, 15% validation and 15% test. (1 point)

3. Training functions. (1 point)

   The training functions should have the following structure:

   - **run**, receives n, the number of epochs and calls the training and validation functions n times, while also updating the training and validation metrics.
   - **train**, trains the model for one complete iteration through the training DataLoader.
   - **val**, validates the model using the validation DataLoader.

   Other structures are also accepted, as long as they are readable, efficient and sensible.

   As seen from the data, the task is one of generating images, so the shape of the output must match that of the target image. Be careful of what output activation function and loss function you choose.

Report training and validation metrics and create a graphic which illustrate their evolution (1 point).
Write an efficient implementation of the training pipeline (resort to caching the images in the dataset class and to other aspects covered during the labs). (1 point)

You must also use at least 4 torchvision transformations (from which 2 have a random factor, aka. augmentations) that will be passed to the dataset class (search for examples in the given documentation).

Your training pipeline must run both on CPU and GPU, using a device parameter.
Upload your homework in the /Lab04/Solution directory. If you are doing your homework in a Jupyter Notebook, add the "Open in Colab" option.

Bonus:

- 1 point for a modular and clear implementation. You can easily change the dataset, transformations, model, and hyper-parameters and the training pipeline remains the same. The Dataset class implementation should be adjusted to become a dataset wrapper that wraps any other custom Dataset class that has to be written when changing datasets.
- 1 point for a ingenious pipeline implementation/structure, that improves efficiency. Must be proven by timing and A/B testing.