Advanced Topics in Machine Learning
Assignment # 1
Universität Bern

Due date: 30/03/2021

1 Introduction

In this assignment you need to upload a zip file to ILIAS which includes: 1) A Jupyter Notebook file Assignment1.ipynb completed with code and answers and 2) a Jupyter Notebook exported to HTML (File / Export Notebook as / HTML). The zip file name must be FirstName LastName.zip. If your implementation requires auxiliary functions, you must implement those functions inside a corresponding .py file. Please state your name at the beginning of the notebook.

1.1 Notes on code and submission quality

In addition to answering the different questions, you are also expected to provide well written submissions. Here are some recommendations to take into consideration.

- Please answer the question in the same order as in the assignment and use the same question numbers.
- Don't answer the questions in the code comments. Use the text cells in your notebook.
- Remove clutter such as unused code lines instead of turning them into comments.
- Make sure the right execution order of the notebook cells is from top to bottom. A TA should be able to reproduce your results by simply clicking "Run All" without having to guess which cells should be executed first.

Poorly written submissions might result in points deduction.

2 Problem

In this assignment, you are asked to predict the mod 10 of sum of two digits on MNIST dataset (see the below figure).

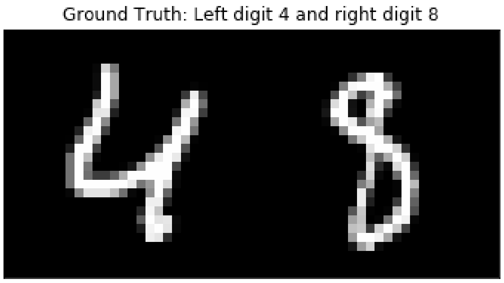


Figure 1: Given these two digit images, your network should predict the following operation: $(4+8) \mod 10 = 2$.

For both training and testing you will use **MNIST-train** and **MNIST-test** datasets including images and their corresponding labels.

Tasks:

1. **Prepare data for the training [Total 25 points]**

   (a) Construct the dataset of pairs of digits from the MNIST dataset. More specifically you should create two datasets i.e. *right* order and *left* order (see the Fig. 2 and the **dataset.py** file given to you) for both training and testing splits of MNIST. Implement a Dataset class that extends Pytorch's Dataset class. Create dataset objects and dataloaders for training and validation datasets. You are provided with a Python file template for that. [**10 points**]

   (b) Set aside 90% of training data for training (for both *right* and *left* order), and the remaining 10% for validation. [**5 points**]

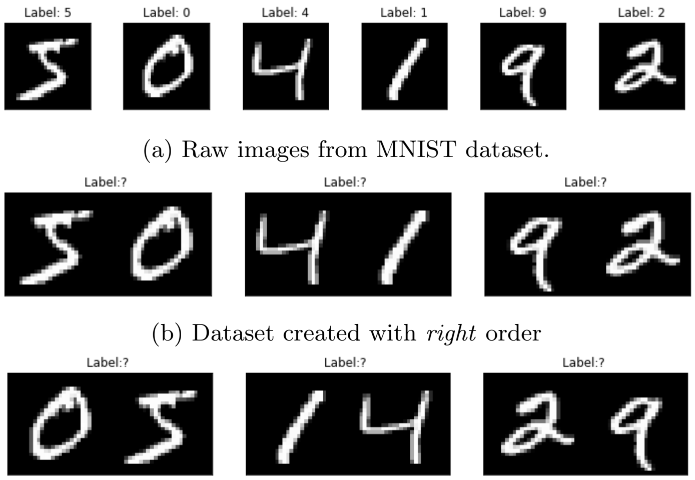
   (c) Concatenate the validation sets (*right* and *left*) into one set and also the test sets into one set using PyTorch library. **You should use these concatenated sets in all your experiments.** [**5 points**]

   (d) Show the label and pair distributions for training, validation and testing sets. [**5 points**]

   Hint: You can visualize the pair distributions using seaborn's heatmap method (use the same range for all cases).

2. **Design training configurations [Total 15 points]**

   (a) Design **at least three** training setups (i.e. models, loss functions and optimizers) using different network configurations (depth, number of hidden neurons) and reasonable learning rates. Your MLP models



(a) Raw images from MNIST dataset.

(b) Dataset created with *right* order

(c) Dataset created with *left* order

Figure 2: Dataset generation process.

should predict the mod 10 of sum of two digits of a given digit pair. The depth of your MLP models should be at maximum 3 layer. [**5 points**]

(b) Implement the training procedure of the model with PyTorch's optimizer given the dataloader as an input. You can find the possible interfaces of the functions in the code templates. You will use this function for the further trainings. [**10 points**]

3. **Training on *right* order dataset** [**Total 15 points**]

   (a) Train the models you implemented on only ***right*** order training dataset. Plot training and validation loss and accuracy per epoch for each model. [**10 points**]

   (b) Select the best model and justify your selection (all model selections should be done on the validation set). [**5 points**]

4. **Training on both *right* and *left* order datasets** [**Total 20 points**]

   (a) Concatenate the *right* and *left* order training datasets. [**5 points**]

   (b) Train the models you implemented on the concatenated dataset. Plot training and validation loss and accuracy per epoch for each model. [**10 points**]

   (c) Select the best model and justify your choice. [**5 points**]

5. **Evaluate on the test set** [**Total 25 points**]

(a) Compare the validation accuracy between previously selected models in points 3 and 4. Select the one with better performance. Explain the difference between the quality of the models trained on different training sets. [**5 points**]

(b) Evaluate the model selected in the previous point on the test set and report your accuracy. Compare this with the validation accuracy. [**5 point**]

(c) Visualize a few samples from your test dataset (both correct and failure cases) for your final model. [**7 points**]

(d) Show the accuracy of your final model on the test set for each digit pair. [**8 points**]

Hint: You can visualize the accuracy using seaborn's heatmap method (use the same range for all cases).