Machine Learning Group
Fakulty IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Milestone 1: Datasets and Prototype

Goal

In this project, your task is to replicate an academic paper by implementing the model and reproducing its results. The aim of this milestone is to become familiar with the dataset and create a prototype. For this, you have to

- get the **data** and implement **visualization tools** that allow you to inspect it (and show examples in your report),
- implement **pre-processing** and **feature extraction** methods if necessary, and
- build a simple **prototype** or **baseline** method.

In this milestone, you are not yet expected to implement the sophisticated aspects of the algorithms. Simple well-chosen baselines/heuristics are sufficient.

Tasks

This list of assignments for this milestone is not exhaustive. Some tasks listed below may not be applicable to your project. If you are unsure about the next steps, it is advisable to communicate with your project supervisor to discuss and align your goals for the initial milestone.

Implementation (1 point)

Data preparation

Write a function

```
def load_data(dataset='<dataset>',
              transformation=None,
              n_train=None, n_test=None, ...)
```

that reads the data from disk, applies some transformations, creates a train/test split, and returns generators for sample/label pairs. Make sure to use *lazy loading*.

- The parameter `dataset` allows to choose a dataset by name, e.g. `dataset='MNIST'`.
- `transformation` is a callable object that is applied to every train and test sample, e.g. re-scaling for images or bag-of-words for text.
- Parameters `n_train` and `n_test` determine the number of train and test samples. They can be `None` for some default split.
- The function may have additional parameters, all of which must have default values.

The output must be a pair of Python generators:

- The first generator returns pairs (x_i, y_i) (or similar, depending on the project) from the **training set**.
- The second generator returns pairs (x_i, y_i) of the same shape, but from a non-overlapping **test set**.

Make sure your data is accessible on the cluster: **store the data in** `/home/space/datasets/<dataset>` **and make it readable** (`chmod -R a+r /home/space/datasets/<dataset>`).

Data visualization

Write a function

```
def show(x, outfile=None, ...)
```

which takes a list of data points and outputs a visualization of the data. The output should be an image (e.g. a contour plot, images arranged in a grid, or rendered text – depending on the kind of data you work with). It should have the ability to output the figure as a file `outfile` when specified. Try to keep the function extensible, as you might want to add overlays later in the project. The function can have any kind of arguments, but make sure that the outputs are visually pleasing (they ought to go in the report).

Model prototype

Write a model class

```
class Model(torch.nn.Module):
    def __init__(self, ...):
        super().__init__()
        [...]

    def forward(self, X):
        [...]
```

that computes the forward pass (i.e. prediction).

- The `__init__` function takes hyperparameters, initializes the model accordingly and sets up trainable parameters (via `torch.nn.Parameter`).
- The method `forward` takes some data `X` and returns the predictions. In the general case, we want the method to be differentiable via automatic differentiation. Thus, make sure to use only `pytorch` functions inside `forward` and that the functions has no conditional statements and loops (it can have these, but you must be careful).

Also implement the loss function and a training loop. For this milestone, the model does not have to converge properly, but you should be able to load data and forward it through the untrained model.

Report (9 points)

Write a report[1] on your work for milestone 1. The following aspects we consider as important and should be included in the report. They are *not* a complete list which you just have to tick off.

Dataset overview (4 points)

- Overview of the data: hierarchy of the targets, number of data points per category. Are there (near-) duplicates, missing values or garbage?
- Describe your feature extraction method (provide literature references if possible), and why you have chosen it.
- Does it make sense to normalize the features in some way? Can the model handle inputs of different shape? What transformations are passed to `load_data`?
- What kind of answers could be provided by machine learning algorithms about the data. What are typical queries a human would ask?

Baseline method and evaluation (3 points)

- Describe a simpler baseline/heuristic that solves the same task.
- How do you evaluate the quality of a prediction? What are meaningful validation metrics for the task?
- Most feature extraction methods and evaluation heuristics are tailored for the binary classification case. Is this an issue here and how do you deal with it?

[1] check the guide on reports on ISIS!

- Report results from the baseline.
- How do you avoid overfitting? Is overfitting an issue?
- Are there significant differences between the categories w.r.t. how well they can be characterized?
- Can something non-technical be said about what the good features are?

Discussion (2 points)

The discussion should summarize your findings.

- What are the challenges of the data set? Why is the task not trivial or why is an answer from the final model actually interesting?
- What does (not) work and why (not)?
- What do you think is possible on the data set? Do you think this will be useful in business?
- Are there things you would like to try but (until now) have not tried?

Literature and Resources

- For the project, you do not have to implement every algorithm on you own. Use `scikit-learn` and `PyTorch` whenever possible.
- `scikit-learn` documentation: `https://scikit-learn.org/stable/documentation.html`
- `PyTorch` documentation: `https://pytorch.org/docs/stable/index.html`

Preliminary Schedule

Lectures, Q&A sessions and presentations take place on Tuesdays, at 10:00.

MS	Topic	Intro	Q&A	Report Deadline	Presentation
1	Data and Prototype	15.10.2024	12.11.2024	22.11.2024, 11:59pm	26.11.2024
2	Implementation and Evaluation	26.11.2024	10.12.2024	03.01.2025, 11:59pm	07.01.2025
3	The Final Method	07.01.2025	21.01.2025	31.01.2025, 11:59pm	04.02.2025

How to submit

- The *report* must be submitted via ISIS. Do **not** include code in your report. Reports that are longer than ten pages will not be accepted. **Use** the LaTeX template provided on the ISIS course page!
- Put the *code* of your project in a folder `MS1` in your home folder on the cluster.
  - The code has be executable on the **cluster**.
  - Do **not** submit jupyter notebooks!