Assignment 1

Deadline: 14.10.2024 – 11.59pm

By the deadline, you are required to submit the following:

Report. Prepare a single PDF file that presents clear and concise evidence of your completion of each of the listed tasks. **Use the overleaf template available on iCorsi**. The report must be **no more than 6 pages**. Pages beyond the sixth will not be reviewed.

Source code. Create a single Python script **using the template available on iCorsi**, performing each of the specified tasks. If a task is accomplished in the code but not documented in the report, it will be considered incomplete. Therefore, make sure to report all your work in the submitted report thoroughly. **Jupyter notebook files will not be accepted**.

General hints. The question marked with * is more challenging, and we recommend possibly leaving it as the last questions to solve. To obtain the maximum grade, not only should all exercises be completed correctly, but the plots must also be clear, have a legend, and have labels on the axes and the text should be written in clear English. For saving plots in high quality, consider using the command `matplotlib.pyplot.savefig`. Clarity of plots and text will account in total for 5 points. Best practices in coding will account for 5 points in total. For this assignment, you don't need GPUs.

Submission rules: As already discussed in class, we will stick to the following rules.

- Use the templates and name your files NAME_SURNAME.pdf and NAME_SURNAME.py (If you have more than one name, just concatenate them). We will read and correct only these files. Other files present in the folder will not be read. If such files are missing, they will be evaluated with a score of 0.
- Code either not written in Python or not using PyTorch receives a grade of 0. Of course you can use auxiliary packages when needed (`matplotlib`, `numpy`,...), but for the learning part, you must use PyTorch.
- If plagiarism is suspected, TAs and I will thoroughly investgate the situation, and we will summon the student for a face-to-face clarification regarding certain answers they provided. In case of plagiarism, a score

reduction will be applied to all the people involved, depending on their level of involvement.

- If extensive usage of AI tools is detected, we will summon the student for a face-to-face clarification regarding certain answers they provided. If the answers are not adequately supported with in-person answers, we will proceed to apply a penalty to the evaluation, ranging from 10% to 100%.

Polynomial regression with gradient descent (90 points)

Let $z \in \mathbf{R}$ and consider the polynomial

$$p(z) = \frac{z^4}{30} - \frac{z^3}{10} + 5z^2 - z + 1 = \sum_{k=0}^{4} z^k w_k \tag{1}$$

where $\boldsymbol{w} = [w_0, w_1, w_2, w_3, w_4]^T = [1, -1, 5, -0.1, \frac{1}{30}]^T$. This polynomial can be also expressed as the dot product of two vectors, namely

$$p(z) = \boldsymbol{w}^T \boldsymbol{x} \quad \boldsymbol{x} = [1, z, z^2, z^3, z^4]^T \tag{2}$$

Consider an independent and identically distributed (i.i.d.) dataset $\mathcal{D} = \{(z_i, y_i)\}_{i=1}^N$, where $y_i = p(z_i) + \varepsilon_i$, and each ε_i is drawn from a normal distribution with mean zero and standard deviation σ.

Now, assuming that the vector $\boldsymbol{w}$ is unknown, linear regression could estimate it using the dot-product form presented in Equation 2. To achieve this we can move to another dataset

$$\mathcal{D}' := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \qquad \boldsymbol{x}^i = [1, z_i, z_i^2, z_i^3, z_i^4]^T$$

The task of this assignment is to perform polynomial regression using gradient descent with PyTorch.

1. (0 pt) Be sure to have installed `numpy 2.1`
2. (5 pts) Define a function

   ```
   plot_polynomial(coeffs, z_range, color='b')
   ```

   Where `coeffs` is a `np.array` containing $[w_0, w_1, w_2, w_3, w_4]^T$, `z_range` is the interval $[z_{\text{min}}, z_{\text{max}}]$ of the z variable; `color` represent a color. Use the function to plot the polynomial in the range $[-4, 4]$. Report the plot. In coding this function, you can use `numpy`.
3. (15 pts) Write a function

   ```
   create_dataset(coeffs, z_range, sample_size, sigma, seed=42)
   ```

that generates the dataset $\mathcal{D}'$. Here `coeffs` contains $[w_0, w_1, w_2, w_3, w_4]^T$, `z_range` is the interval $[z_{\text{min}}, z_{\text{max}}]$ of the z variable; `sample_size` is the dimension of the sample; `sigma` is the standard deviation of the normal distrbution from which ε_i are sampled; `seed` is the seed for the random procedure. Your function must return two `torch.tensor` data with the noisy dataset.

4. (5 pts) Use the code of the previous point to generate data with the following parameters
   - Each z_i should be in the interval $[-2, 2]$
   - $\sigma = 0.5$
   - Use a sample size of 500 for training data and a seed of 0
   - Use a sample size of 500 for evaluation data and a seed of 1
5. (10 pts) Define a function

   ```
   visualize_data(X, y, coeffs, z_range, title="")
   ```

   that plots the polynomial $p(z)$ and the generated data, (train and validation), where `X, y` are as returned from the function `create_dataset`, `coeffs` are the coefficient of the polynomial, `z_range` is the interval $[z_{\text{min}}, z_{\text{max}}]$ of the z variable and `title` may be helpful to distinguish between the training and the validation plots. More specifically: Provide two plots containing both the true polynomial; In one, add a scatter plot with the training data, and in the other a scatter plot with the validation data. Use the function to visualize the data. At this point, you should have noticed that, depending on the circumstances, it is advantageous to store the polynomial coefficients either from the degree 0 term to the degree n term, or vice versa.
6. (30 pts) Perform polynomial regression on $\mathcal{D}$ using linear regression on $\mathcal{D}'$ and reports some comments on the training procedure. Explain and comment on the training procedure, focusing specifically on:
   - (15 pts) Which learning rate do you use, and why; What happens if the learning rate is too small; what happens if the learning rate is too high, and why?
   - (5 pts) If `bias` should be set as True of False in `torch.nn.Linear` and why.
   - (10 pts) How many steps did you use and why did you retain they were enough?

   You should use the data you generated at point 4. **Note:** If you plan to reuse code explained during the lecture, explicitly indicate which parts are being reused or readjusted/modified. In any case, the steps must always be explained and understood in detail.

7. (10 pts) Plot the training and validation loss as functions of the iterations and report them in the same plot. If you use both steps and epochs, you can choose either of the two, as long as it is clear from the plot and the plot reports what we expect - namely, that the loss functions decrease.
8. (5 pts) Plot the polynomial you got with your estimated coefficients as well as the original one in the same plot.
9. (10 pts) Plot the value of the parameters at each iteration as well as the true value. What we expect is a plot having on the x axis the number of steps/epochs and on the y axis the value of the parameters while they are learned. Then, in the same graph, plot a horizontal line with the true value.

Question (5 pts)

Explain the difference between choosing two different learning rates *in principle* and what is instead done by the adaptive methods - e.g, Adagrad. If you don't know the method, you can study one of the following resources:

- Section 8.5.1 of Deep Learning Book
- Section 12.7 of Dive Into Deep Learning

Bonus question* (5pts)

Suppose you have to learn the function

$$f(x) = 2\log(x+1) + 3$$

and you know that your data are x^i belong to the interval $[-0.05, a]$. Which performances would you expect using linear regression in the following cases?

1. $a = 0.01$
2. $a = 10$

Note: We want to see some code that generates noisy data from f, do the linear regression and we want to see the loss value on it. If you know the mathematics behind it, you can use your knowledge to answer and validate the data, but as this is a Lab exam, we want to see empirical evidence.