Due date: October 29, 2024
Please submit a pdf with answers (add figures if appropriate)
as well as the notebook used to generate them.

1 Linear regression with pytorch (10 pt)

The jupyter notebook `linear-regresssion.ipynb` contains an example of a linear regression. The linear model is implemented using the `pytorch` package as a linear neural network layer with one weight and one bias parameter. The steps include:

1. creating a dataset
2. randomly dividing the dataset into test and training sets
3. fitting using the training set
4. evaluation of the predictive power

Go through the notebook and familiarize yourself with the different steps. In case the functionality of any of the imported `pytorch` functions is not clear to you, additional information can be found in the respective documentation (https://pytorch.org/docs/stable/index.html).

1.1 Linear model

1.1.1 Data generation

Random numbers are used to create the dataset which is then split into test and training sets. Explain what the function `np.random.randn()` does and how it is used in the workflow.

1.1.2 Datasets and DataLoaders

Have a look at https://pytorch.org/tutorials/beginner/basics/data_tutorial.html. Briefly explain the concepts of `Dataset` and `DataLoader`.

1.1.3 Building and training a model

Write down the mathematical expression for linear model used in the regression. What are the trainable parameters? How are the parameters that correspond to the weights retrieved in the `pytorch` model? Which function is used to make predictions with the fitted model?

1.1.4 Model evaluation

To evaluate the predictive power of the model, we evaluate the mean squared error (MSE), the mean absolute error (MAE) and the coefficient of determination R^2. The function `r2_score` returns the coefficient of determination R^2. Write down the equation to compute R^2. What does a value of $R^2 = 1$ mean? What does a value of $R^2 = 0$ mean?

1.2 Experiments

Once you are familiar with the all the steps explained above, you can use the code blocks to do the following experiments.

1.2.1 Different datasets

Create 3 different datasets with $d_{\max} = \{1, 10, 20\}$ and fit a linear regression model for each of these datasets. Compute R^2, MAE, and MSE for each fit. How do the values change as a function of $d_{\max}$? Explain the observed trend.

1.2.2 Scaled data

Create a dataset over a larger interval, e.g. with `high_val` = 50 or 100 and try to train a model. What do you observe? What could be the reason? Try with a (much) smaller learning rate.

To fix the occuring problem, the data needs to be scaled. To do so, we can use the class `PTMinMaxScaler` implemented in the notebook which scales all the data into a given interval (for training this interval should be chosen to be [0,1]). Use this class to scale all X and y values and train the model again. Compute the MAE and MSE (don't forget to transform all y values back to the original scale before the evaluation – this can be done using the function `inverse_transform` from the scaler class.)