Heidelberg University
Institute of Computer Engineering
Computing Systems Group

Holger Fröning
Hendrik Borras
Christian Simonides

Embedded Machine Learning
Summer Term 2024

Exercise 1

- **Return electronically until Wednesday, April 24, 2024, 09:00**
- **Include your names on the top sheet. Hand in only one PDF.**
- **A maximum of four students are allowed to work jointly on the exercises.**
- **When you include plots add a short explanation of what you expected and observed.**
- **Hand in source code if the exercise required programming. You can bundle the source code along with the PDF in a .zip file.**
- **Programming exercises can only be graded if they run on the cluster in the provided conda enviroment. Make sure to document additional steps, which you might have taken to run the exercises.**

1.1 Reading

Read the following paper and provide a review as explained in the first lecture (see slides):

- Colby Banbury, et al. 2021. MLPerf Tiny Benchmark.

(5 points)

1.2 Polynomial curve fitting

This exercise is meant as a gentle introduction to Python and the issue of overfitting.

- Install numpy on your machine, if you don't have Python (and/or work in Windows) it is easiest to install anaconda, see `https://docs.conda.io/projects/conda/en/latest/index.html`. If you have Python installed you can use pip `pip install numpy torch matplotlib`
- Start with the template `template.py` in the ,,template/" folder. The file should already produce two plots, with as of yet incorrect data.
- Extend the `ground_truth_function` to generate data of the shape $h(x) = sin(2\pi x)$, as in the lecture.
- Extend the `error_function` to reflect the non-regularized error function from the lecture.
- For this exercise we will be using the `numpy` function `polynomial.Polynomial.fit` to fit the polynomial for us. If everything is setup correctly you should by now get a plot for a polynomial fit of third degree, similar to the one shown in the lecture.
- Create a new plot for an overfitted Polynomial of 11-th degree.
- Now vary the degree of the Polynomial from 0 to 11. With this data reproduce the plot of "Polynomial degree against the train and test error" from the lecture. Make use of the already created data and make sure to use the RMS-Error instead of the direct error.
- Now vary the size of the data, but keep the degree of the Polynomial constant. As a sensible starting point choose a 10-th degree polynomial. Then vary the sample size of the dataset from 10 until the RMS-Error of the train and test dataset are of similar size, i.e. the function doesn't over fit anymore. Note that the required sample size can easily be of the order of a few 100 for this to fully happen.

(20 points)

Holger Fröning
Hendrik Borras
Christian Simonides

1.3 Willingness to present

Please declare whether you are willing to present any of the exercises from this sheet. The declaration should happen on the PDF which you hand in.
The declaration can be made on a per-exercise basis. Each declaration corresponds to 50% of the exercise points. You can only declare your willingness to present exercises for which you also hand in a solution. If no willingness to present is declared you may still be required to present an exercise for which your group has handed in a solution. This may happen if as example nobody else has declared their willingness to present.

- Reading (Section: 1.1)
- Polynomial curve fitting (Section: 1.2)

(13 points)

Total: 38 points