Uncertainty in Machine Learning - WBAI054-05.2021-2022.2B

Assignment 1

Total points: **40**
Starting date: 11 May 2022
Submission deadline: **23:59, 24 May 2022**

Lecturer Dr. Matias Valdenegro
Teaching Assistants Peter Varga and Cosmin Roger Harsulescu.
General guidelines:

- The tasks are targeted at groups of three students. Please make sure that the load is well divided: every student should contribute. In the case of not equal contributions that grade will be divided accordingly.
- Please take advantage of the tutorial sessions to ask your questions about the tasks.
- Provide a (short but comprehensive) explanation of what you are doing for each task.
- A reviewer should be able to understand plots independently; be sure to label axes, a legend for colors, use an easily readable font size, etc.
- Refer to all plots, tables, code blocks, etc. in your report.
- Submit the report as a single PDF file through Nestor.
- Submit all code in a zipped file named *Code_Group_GroupNumber*, e.g., *Code_Group_20* through Nestor.

Part I - Sources of Uncertainty

Maximum obtainable points: **12**

In our first theoretical lecture we discussed the two sources of uncertainty, aleatoric and epistemic, and included some examples. In this exercise, please produce two new examples (different from the ones in the lecture) for each of epistemic and aleatoric uncertainty (four examples in total), that showcase the corresponding concept.

Part II - Are They Probabilities?

Maximum obtainable points: **12**

Let's compare DUQ and Gradient Uncertainty, both presented in the Lecture *Uncertainty Quantification Methods I.*
Both of them use unconventional ways to produce a prediction with some confidence. DUQ uses a radial basis function, and Gradient Uncertainty uses the gradient of the classification loss. Discuss and answer the following question:

Are the confidence outputs produced by DUQ and Gradient Uncertainty considered as probabilities?

Part III - Programming

Maximum obtainable points: **16**

Generate a small regression dataset by sampling a non-linear mathematical function of your choice (sinusoids, exponential, logarithm, etc), complemented with Gaussian noise with parameters $\mu = 0$ and σ of your choice that makes sense. Train a neural network with two output heads, one for the mean and another for the variance, using the Gaussian Negative Log-Likelihood loss.

Hint: As a source of motivation for implementing the loss function, you are allowed to use the code available at:
`https://github.com/mvaldenegro/keras-uncertainty/blob/master/keras_uncertainty/losses.py`

Discuss your results, plot your dataset, and the predictions of your model, particularly analyze if the variance head of the model is estimating aleatoric uncertainty (your Gaussian noise added to the function) correctly or not.