Course Challenge

Adityanarayanan Radhakrishnan

January 14, 2023

Challenge description. The goal of this challenge is to understand how to approach machine learning model development given data and to learn to incrementally build towards better solutions to a difficult problem. For this challenge, I have picked a particular, nontrivial data distribution and have given you access to samples (features, X, and targets, y) from this distribution. In particular, I have uploaded the following six files:

1. `X_train.csv` and `y_train.csv`. These contain the training data for this challenge. The features for X_{train} should be of shape 5000×50, and the targets y_{train} should be of shape 5000×1.
2. `X_val.csv` and `y_val.csv`. These contain the validation data for this challenge. The features for X_{val} should be of shape 1000×50, and the targets y_{val} should be of shape 1000×1.
3. `X_test.csv` and `y_test.csv`. These contain the test data for this challenge. The features for X_{test} should be of shape 5000×50, and the targets y_{test} should be of shape 5000×1.

I have also uploaded code to load data from these files using `numpy` (see `competition_code.py`).

Challenge goals. There are three goals to this challenge, which are in line with the goals of this course.

1. **Performance.** The first goal is to achieve the best R^2 on the test samples when building a machine learning model on the training data. The purpose of validation data is for hyper-parameter tuning. In particular, if you are grid searching over hyper-parameters for kernel regression (e.g. ridge regularization or bandwidth) or over architectures for neural networks, you should select the best model based on validation R^2 and then report the performance of these parameters on the test set. Your submission will not count if you select hyper-parameters based on the test data.
2. **Interpretability.** It is possible to get above .95 test R^2 on this problem. If you do so, you should be able to mathematically explain how your model got this performance. An example of interpretability would be explaining which coordinates are most relevant for prediction and presenting some experiments verifying your explanation.
3. **Simplicity.** While you are of course free to use specialized resources if these are available to you, submissions that do not used specialized computing resources will receive a higher rating than those using these resources. In particular, it is possible to get .95 test R^2 with a laptop using non-GPU software. Simplicity is particularly important for making sure that the tool you used to get your good result is accessible to as broad a community as possible.

Challenge scoring. If you get an R^2 above .95, then you get 1 point. If you get the highest scoring model, you get 2 points. If your model gets above .95 R^2 and you are able to mathematically explain why your model performs well, you get 1 additional point. If your model gets above .95 R^2 and does not use specialized hardware (e.g. GPUs), you get 1 point.

Challenge submissions. There will be an assignment lasting until Jan. 27 allowing you to submit your submissions. For this submission, you will need to report the following along with providing your code:

1. The training, validation, and test mean squared error and R^2 for your model.

2. A brief description of which model you used and which hyper-parameters.
3. A description of why your model performed well and in particular, which features were relevant.
4. A description of the hardware and software used to train your model.

Challenge rewards. Everyone is highly encouraged to participate in this challenge. The model that scores highest according to the scoring system above will have the opportunity to present their model on the last day of the course and gets full credit for the project proposal/paper review and the last problem set. Ties will be broken by order of submission.