IWLS 2021 Programming Contest: ML+LS (Part II)

Alan Mishchenko (UC Berkeley) and Satrajit Chatterjee (Google AI) and Gai Liu (Xilinx)
iwls.contest.2021@gmail.com

Introduction. The contest this year extends, in a fairly significant way, the contest from last year. The goal last year was to learn an unknown boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ from a training set consisting of input-output pairs. The contest generated a large number of interesting ideas and approaches that are summarized in this paper [DATE 21]. This year, we extend it to the multi-output case, that is, to the case of learning a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$.

Since this is much more challenging than the task last year, instead of 100 separate benchmarks, we will use only one benchmark dataset: CIFAR-10.

Task. The goal is to learn an and-inverter graph (AIG) with a limit on the number of internal and-nodes, which performs well on the CIFAR-10 dataset. Since each CIFAR-10 image is 32x32 pixels with three 8-bit color channels, the AIG has 32*32*24=24,576 binary inputs. The ordering of the binary inputs is the same as that of the bits in the binary version of CIFAR-10 dataset. Since there are 10 output classes, the AIG has 10 binary outputs. Thus, the function to be learned is $f : \{0, 1\}^{24576} \rightarrow \{0, 1\}^{10}$. (The input/output nodes do not count as internal and-nodes.)

We consider 3 different size limits: `small` (where the AIG has no more than 10,000 and-nodes), `medium` (no more than 100,000 and-nodes), and `large` (no more than 1,000,000 and-nodes). Each category will have a separate winner. (Note that in the `small` category, there is not even enough budget to incorporate all the input information! So it is expected that some input nodes in the resulting AIG may have no fanout.)

Evaluation. The score will be based on the Top-1 accuracy on the public CIFAR-10 **test set** (please see the rule below for what you can and cannot use for training). However, the exact metric will be decided later. ABC has command `&iwls21test` to check the AIG parameters and the resulting accuracy, for example: `&iwls21test small.aig data_batch_1.bin`

(The output values produced by the AIG are expected to be one-hot for each input combination. If multiple outputs for the same input combination have value one, the resulting class is assumed to correspond to the lowest asserted output. If no output is asserted, class 0 is assumed.)

What to submit. Please submit 3 AIG files in Binary AIGER format named `small.aig`, `medium.aig`, and `large.aig`. Please also submit your code and a short write-up on the method used. At the top of the write-up, please indicate how many times you peeked at the CIFAR-10 test set in the course of your research. (There may be a special prize for 0 or 1 peeks.)

Very Important Rule. Since CIFAR-10 is a public dataset, we will rely on the honor system to evaluate generalization. **Please do NOT use the CIFAR-10 test set (`test_batch.bin`) for training.** Please also do not use the test set for extensive hyperparameter tuning. You may be occasionally tempted to see how you are doing on the test set, but each time you do so, please record it. As noted above, **when submitting your entry, please indicate how many times you have looked at the CIFAR-10 test set in the course of the project.**