Tuning the Learning of Circuit-Based Classifiers

Master thesis

Bernhard Gstrein

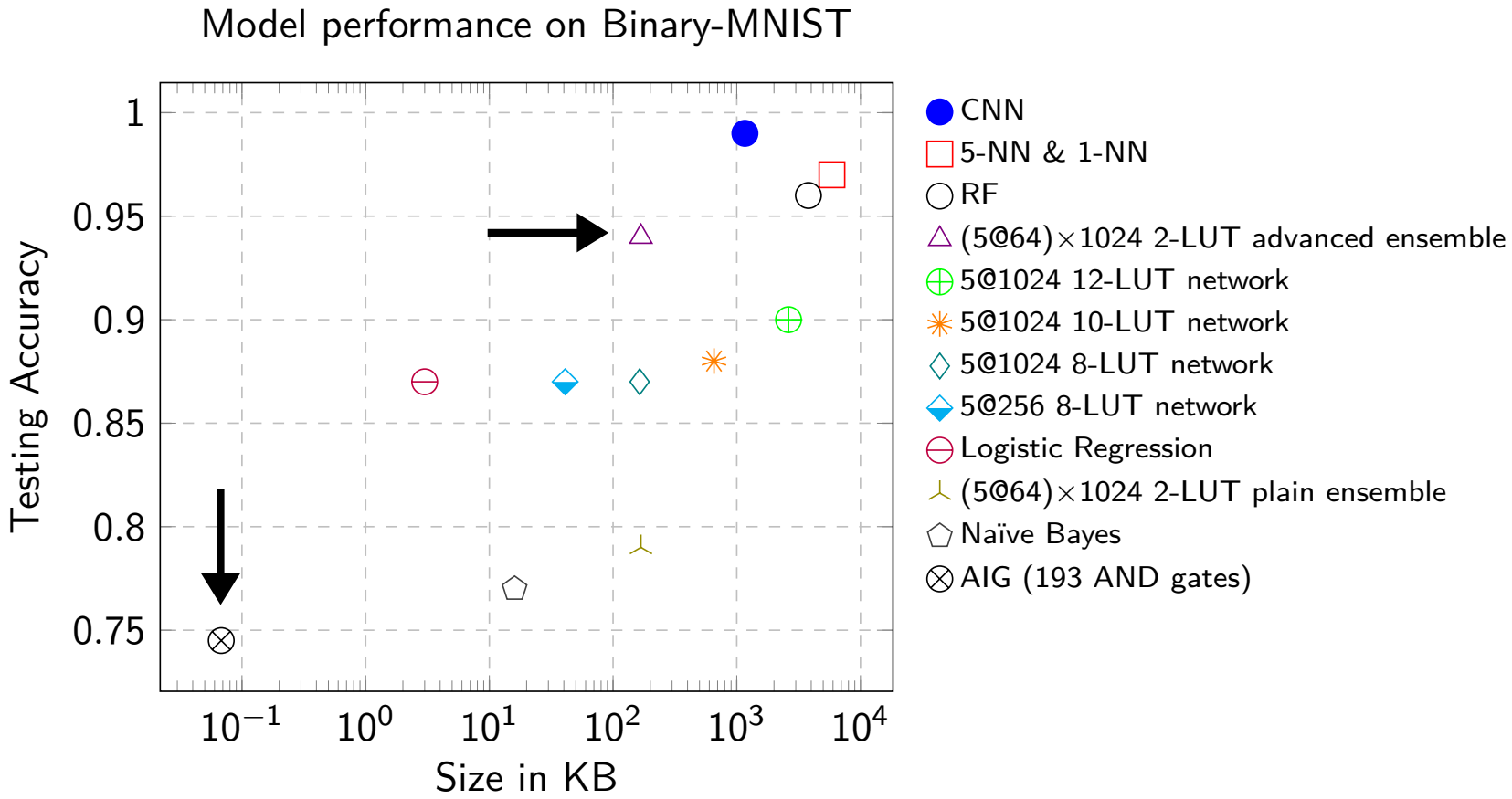
Motivation

- Neural Networks blow our mind, but often have lots of parameters
- High computational costs for training and inference
  - Environmental impact is a concern [1]
  - Need to deploy on cloud (and thus, internet connection)
- Circuit-based models could be a step toward efficiency
- Construct classifiers that work internally with binary variables

Approach

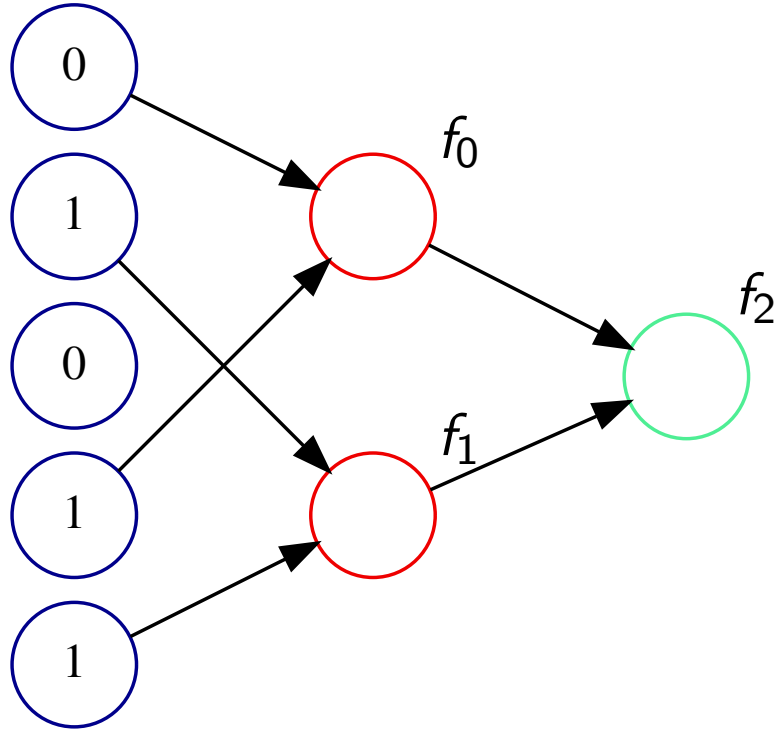
- Most common: gradually shrink neural architectures [2]
- We already start at binary
- Lookup table (LUT) networks [3], [4] and LUT learning algorithm
- And-inverter graphs (AIGs) and local search

Results


Model performance on Binary-MNIST
Testing Accuracy
1
0.95
0.9
0.85
0.8
0.75
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
Size in KB
CNN
5-NN & 1-NN
RF
(5@64)×1024 2-LUT advanced ensemble
5@1024 12-LUT network
5@1024 10-LUT network
5@1024 8-LUT network
5@256 8-LUT network
Logistic Regression
(5@64)×1024 2-LUT plain ensemble
Naïve Bayes
AIG (193 AND gates)

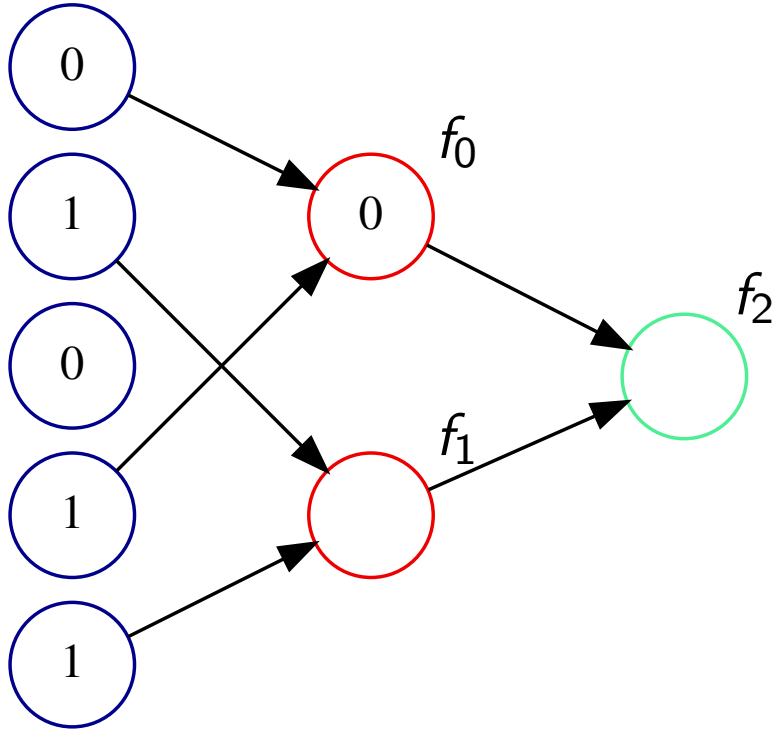
Simple LUT Network

bit-pattern	f_0	f_1	f_2
00	1	1	1
01	0	0	1
10	1	0	0
11	0	1	0



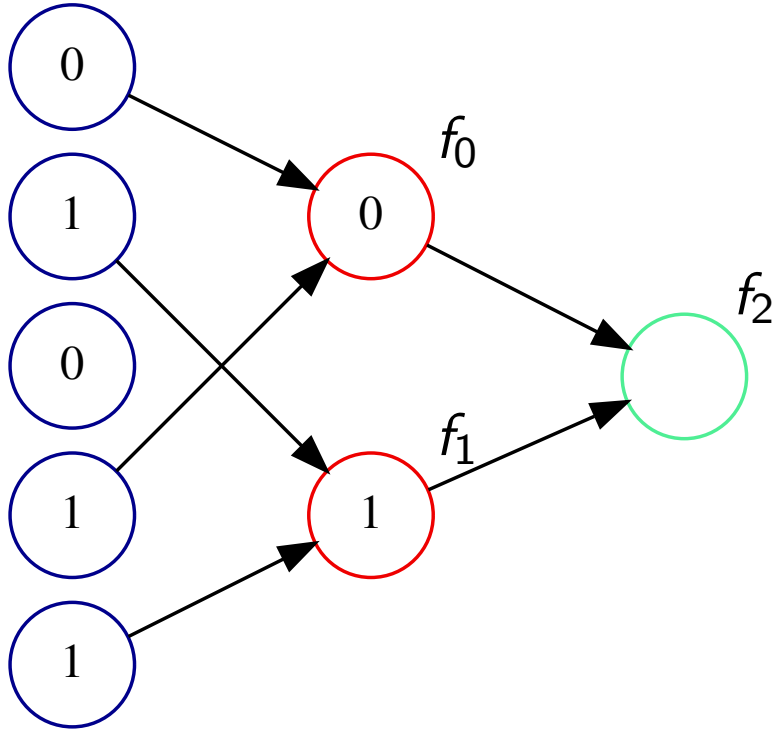
Simple LUT Network

bit-pattern	f_0	f_1	f_2
00	1	1	1
01	0	0	1
10	1	0	0
11	0	1	0



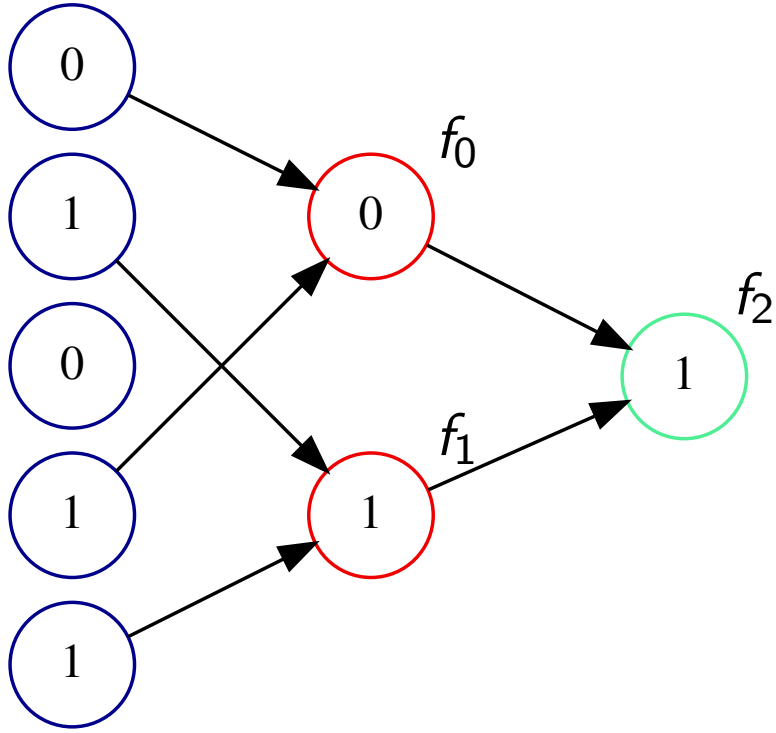
Simple LUT Network

bit-pattern	f_0	f_1	f_2
00	1	1	1
01	0	0	1
10	1	0	0
11	0	1	0



Simple LUT Network

bit-pattern	f_0	f_1	f_2
00	1	1	1
01	0	0	1
10	1	0	0
11	0	1	0



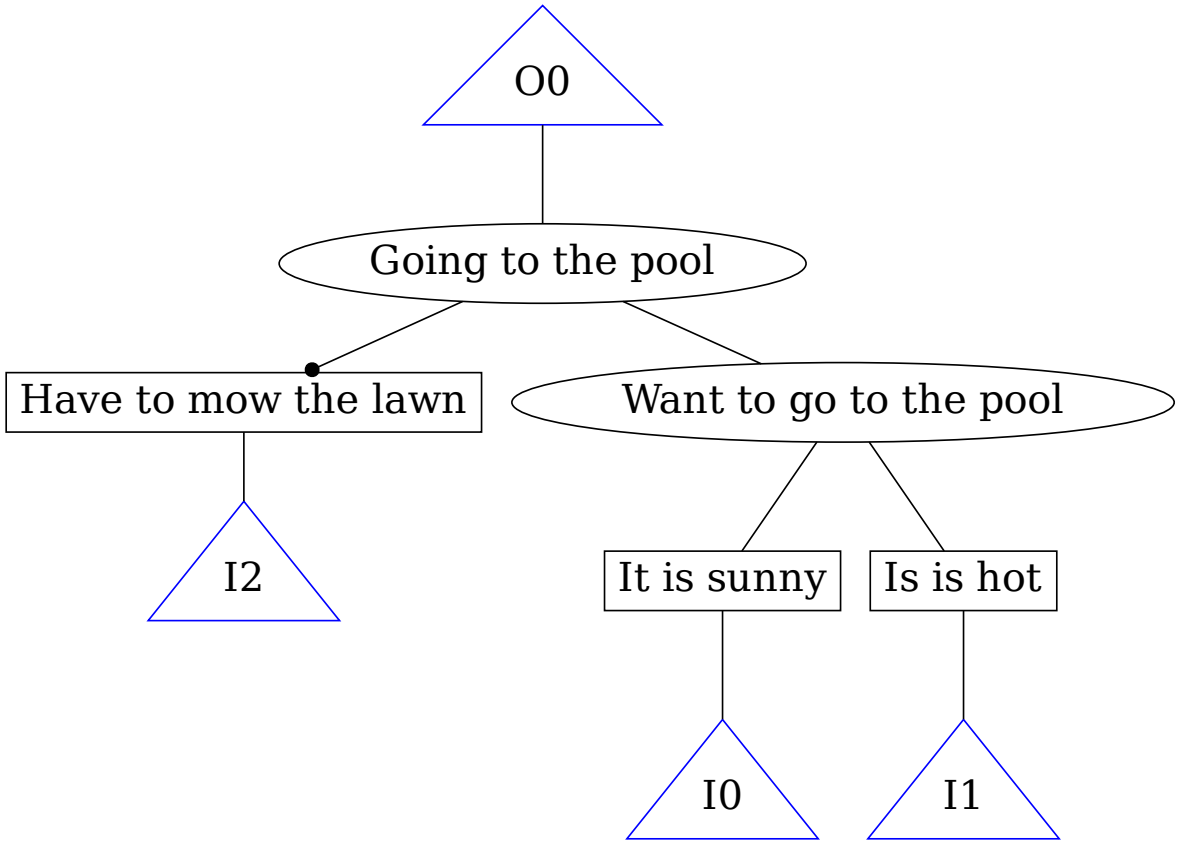
LUT networks: Our Contributions

- Wrote Python code from scratch in approximately 600 LOC
  - Used NumPy library extensively [5]
- Recreated one experiment from [3], got the same results
- Taking the idea of LUT networks further:
  - Modify existing LUT networks or the LUT network learning algorithm
  - Enhance the dataset using feature engineering
  - Combine many LUT networks of low arity (ensembling)

LUT networks: Findings

- Small tweaks boosted the performance of low-arity LUT networks
- Feature engineering using Sobel-inspired operator gave minor improvement
- AdaBoost.M1-inspired ensemble had reduced size and gave best accuracy yet (90% $\rightarrow$ 94%)

AIG Classifier



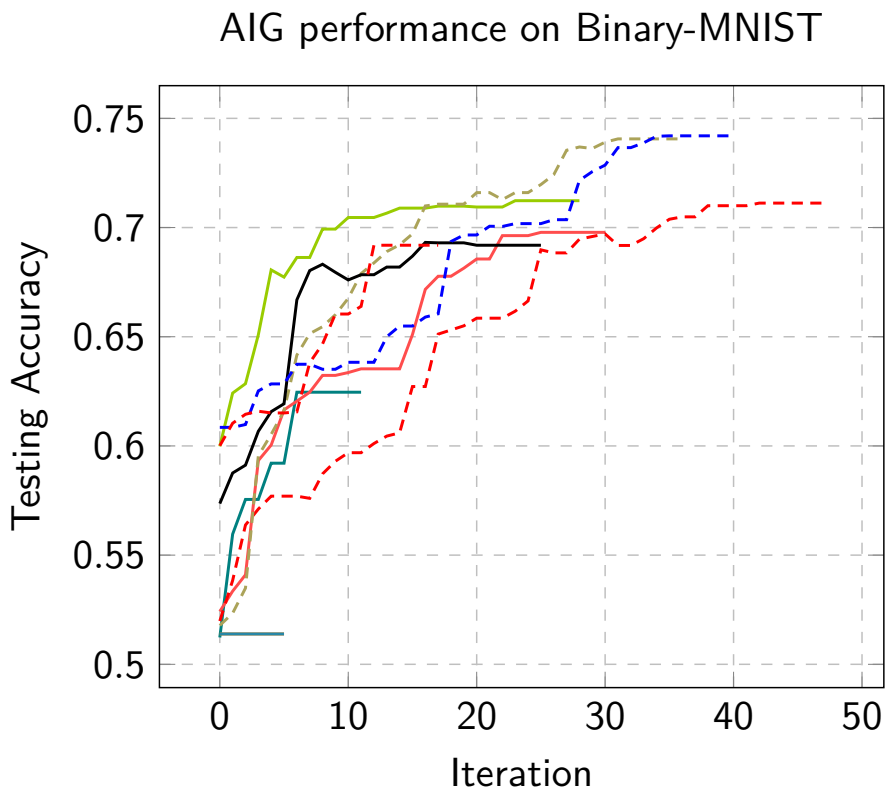
$$(\neg I2) \wedge (I0 \wedge I1)$$

AIG Classifier: Our Contributions

- Usually, AIGs used as intermediate representations for circuits
- Our idea: use AIGs as classifiers
- Devised an initialization scheme and a local search algorithm
- Wrote C++ code from scratch in approximately 900 LOC
  - AIG structure, local search, export to other formats
  - AIGER library [6] used for inference

AIG Classifier: Results

- We conducted 42 experiments
- The resulting AIGs were tiny
- Implementation on edge devices is promising



Conclusion

- Topic of this thesis: building binary predictors and starting from binary
- LUT: Recreated paper [3], got the same results
- Improved LUT network accuracy while reducing model size
- AIG: Devised initialization and local search scheme
- Tiny AIG size is promising for implementation on edge devices

References I

[1] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[2] D. Liu, H. Kong, X. Luo, W. Liu, and R. Subramaniam, "Bringing AI to edge: From deep learning's perspective," *Neurocomputing*, 2021.

[3] S. Chatterjee, "Learning and memorization," in *International conference on machine learning*, 2018, pp. 755–763.

[4] S. Chatterjee and A. Mishchenko, "Circuit-based intrinsic methods to detect overfitting," in *Proceedings of the 37th international conference on machine learning, ICML 2020, 13-18 july 2020, virtual event*, 2020, vol. 119, pp. 1459–1468. Available: http://proceedings.mlr.press/v119/chatterjee20a.html

References II

[5] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[6] A. Biere, "The AIGER And-Inverter Graph (AIG) format version 20071012," Institute for Formal Models; Verification, Johannes Kepler University, Altenbergerstr. 69, 4040 Linz, Austria, 07/1, 2007.