1 Summary

A Hierarchical Neural Model of Data Prefetching (2021):

- Voyager is a neural network for data prefetching that can learn delta correlations and address correlations, i.e. this one can perform temporal prefetching, which was not previously done in other models, as rule-based prefetchers are limited by pre-determined strides and fixed methods of correlating addressed, and other ML-based prefetchers do not perform well for irregular data prefetches or lacked sufficient measurements that consider practical accuracy and timeliness.
- Class explosion problem (too large input and output space) addressed by decomposing address prediction into 1) page prediction and 2) offset prediction, and using an attention-based embedding layer so that pade prediction can provide context for offset prediction.
- Labeling problem addressed by using multi-label training scheme for model to learn from multiple possible labels and it learns the most predictable label.
- Neural models not yet practical for use in hardware data prefetchers.
- Their paper is first to 1) show IPC benefit of LSTM prefetching, 2) show a meural model that combines both delta patterns and address correlations, and 3) their multi-labeling scheme provides a richer set of labels, 4) their model is more compact and less computationally expensive than prior neural solutions.

2 Strengths

-

3 Weaknesses

-

4 Rating: 4

5 Comments

6 Notes

- Voyager (2021):

  - a new neural network for data prefetching. A practical neural prefetcher. Probabilistic model of data prefetching.
  - can learn **address correlations**/temporal prefetching (for prefetching irregular sequences of memory accesses). It can also accomodate **delta correlations**/patterns (strides).
  - has a hierarchical structure separating addresses into pages and offsets, which introduces mechanisms for learning relations among pages and offsets. The hierarchical treatment of data addresses helps accomodate address correlations.
  - SPEC 2006 and GAP benchmark suites (irregular SPEC and graph): 41.6% IPC improvement over system with no prefetching, 21.7% IPC improvement over Domino prefetcher, 28.2% IPC improvement over ISB prefetcher. It has 79.6% accuracy/coverage of the benchmarks.
  - lower overhead: 15-20x reduced computation (training and prediction) cost, 110-20x reduced storage overhead (model size, Voyager is also smaller than non-neural temporal prefetchers); normal neural models not in hardware due to slow training and prediction.
  - They also showed the prefetching results on Google search and Google ads, Voyager achieves 37.8% and 57.5% accuracy/coverage, respectively.

- Neural Prefetchers:
  - they still have a lot of computational cost that makes them impractical for hardware
  - authors found long data address histories is a good feature to predict irregular accesses.
  - multiple localizers benefit some hard to predict benchmarks.
  - These insights are meant to to guide development of practical prefetchers.

- Data prefetching problems:
  - class explosion problem: data prefetching has enormous inputs and output spaces, i.e. for 64-bit address address space, a model needs to predict from among 2^{64} unique address values.
    * authors address Class explosion problem addressed by decomposing address prediction into 1) page prediction (space is 10s-100s thousands) and 2) offset prediction (space is 64).
    * offset aliasing problem: addresses with same offset will share same offset embedding (internal representation of input features in a neural network, learned during training such that features that behave similarly have same embeddings.), leading to poor performance in neural networks. The authors use a new attention-based embedding layer that allows page prediction to provide context for offset prediction.

- labeling problem: data prefetchers have no known ground truth tables from which to learn. Its not clear which label to use to train the ML model.
  * branch predictors can be trained by the ground truth answers as revealed by program's execution.
  * cache replacement policies can be trained by learning from Belady's provably optimal MIN policy.
  * to address labeling problem, authors use a new form of localization built into Voyager, a multi-label training scheme, enabling model to learn from multiple possible labels. (no single ground truth table, but model learns the most predictable label)

•

References

[1] Zhan Shi, Akanksha Jain, Kevin Swersky, Milad Hashemi, Parthasarathy Ranganathan, and Calvin Lin. 2021. A hierarchical neural model of data prefetching. Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. Association for Computing Machinery, New York, NY, USA, 861-873. DOI:https://doi.org/10.1145/3445814.3446752