NeuroVectorizer: End-to-End Vectorization with Deep Reinforcement Learning

paper review

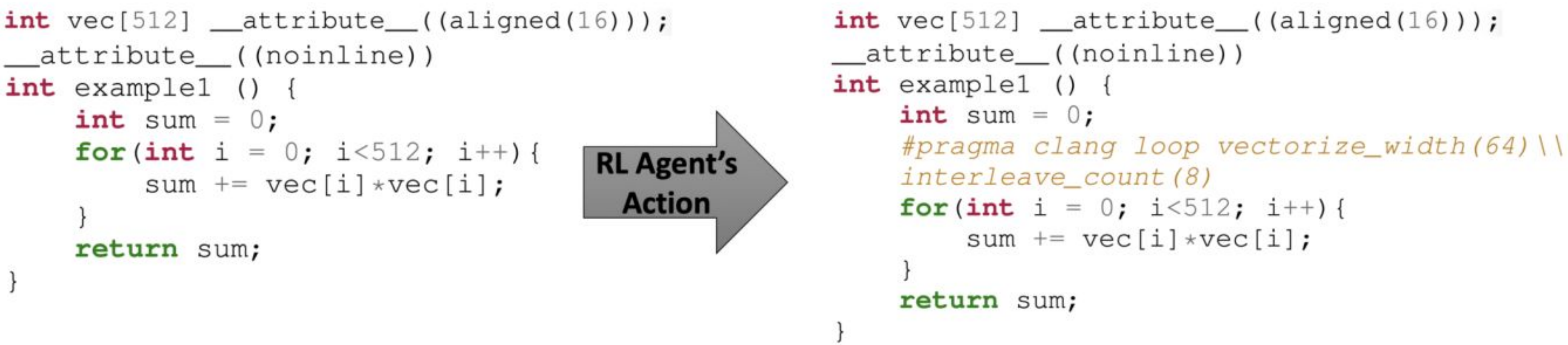
Bagryansky Lev

Meta data

- **Authors**: Ameer Haj-Ali, Nesreen K. Ahmed,Ted Willke,Yakun Sophia Shao, Krste Asanovic, Ion Stoica - USA, Berkeley.
- **Conference:** CGO 2020: Proceedings of the 18th ACM/IEEE International Symposium on Code Generation and Optimization.
- **Year:** 2020.
- **Citations:** 37.
- **References:** 40.
- **Downloads:** 2721.
- **Pages:** 14.

What is the paper about?

They have solved the problem of vectorization through cycles. They suggest several solutions using neural networks.

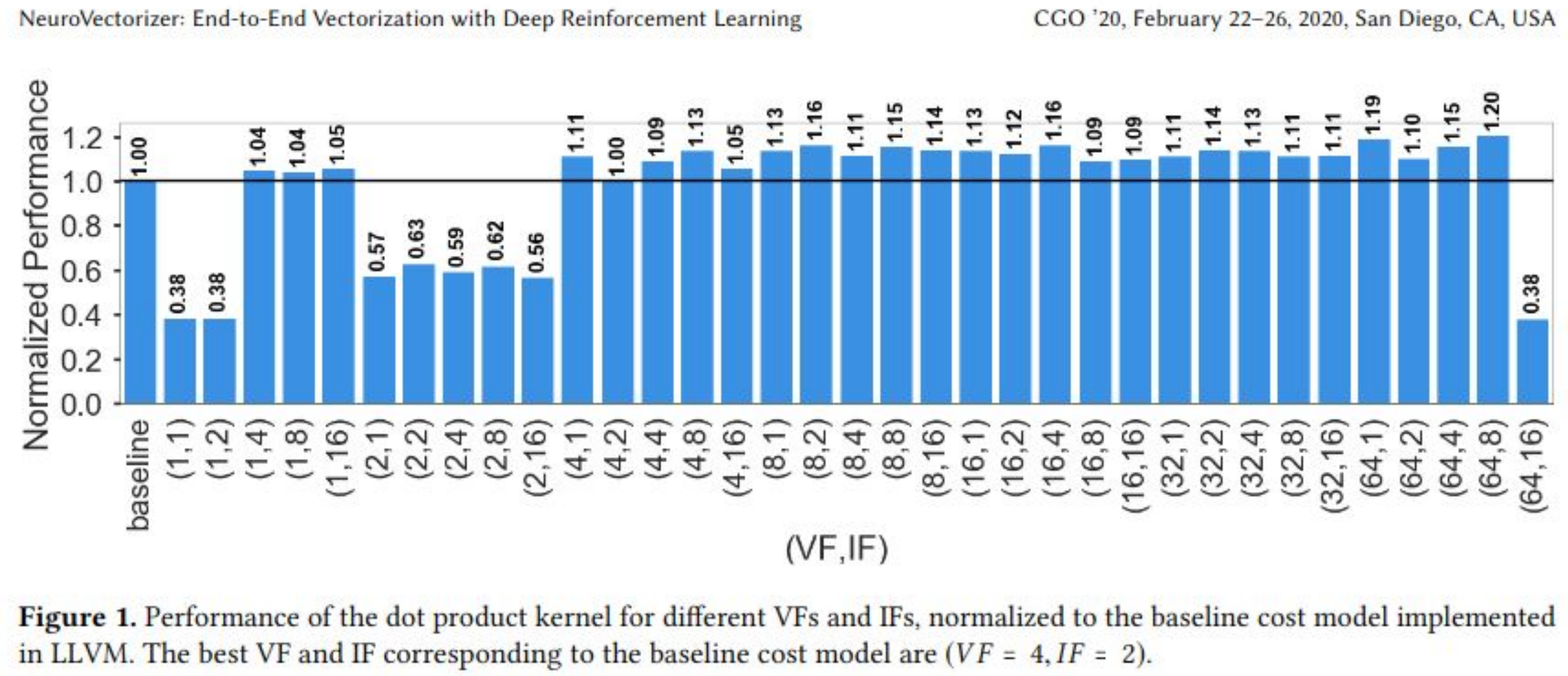


```
int vec[512] __attribute__((aligned(16)));
__attribute__((noinline))
int example1 () {
     int sum = 0;
     for(int i = 0; i<512; i++){
          sum += vec[i]*vec[i];
     }
     return sum;
}
```

```
int vec[512] __attribute__((aligned(16)));
__attribute__((noinline))
int example1 () {
     int sum = 0;
     #pragma clang loop vectorize_width(64)\\
     interleave_count(8)
     for(int i = 0; i<512; i++){
          sum += vec[i]*vec[i];
     }
     return sum;
}
```

Figure 4. An example of the automatically injected VF and IF pragmas by the RL agent.

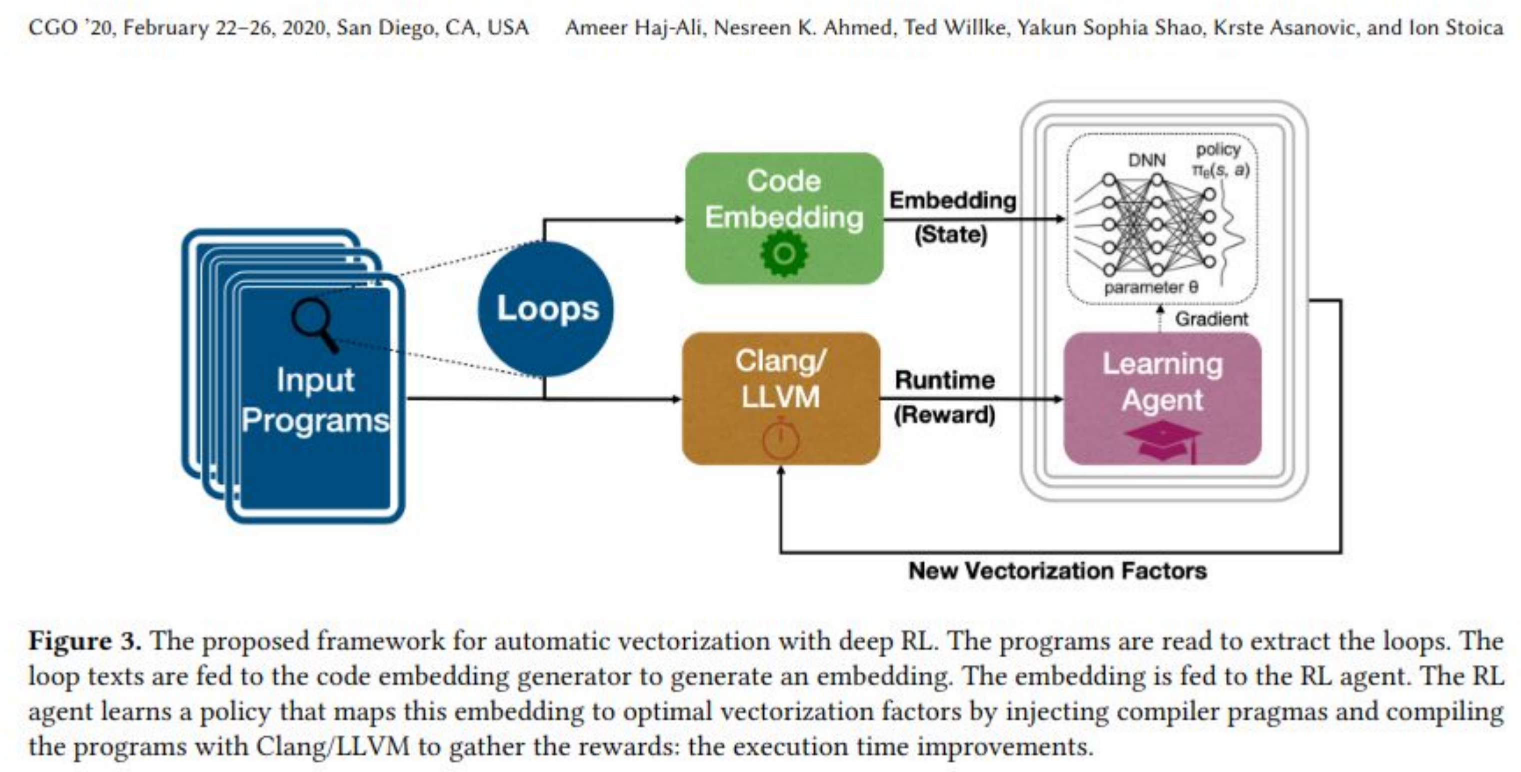
What is the paper about?



NeuroVectorizer: End-to-End Vectorization with Deep Reinforcement Learning CGO '20, February 22–26, 2020, San Diego, CA, USA

Figure 1. Performance of the dot product kernel for different VFs and IFs, normalized to the baseline cost model implemented in LLVM. The best VF and IF corresponding to the baseline cost model are ($VF = 4, IF = 2$).

What is the paper about?



CGO '20, February 22–26, 2020, San Diego, CA, USA Ameer Haj-Ali, Nesreen K. Ahmed, Ted Willke, Yakun Sophia Shao, Krste Asanovic, and Ion Stoica

Input Programs
Loops
Code Embedding
Embedding (State)
DNN
policy $\pi_\theta(s, a)$
parameter θ
Gradient
Clang/ LLVM
Runtime (Reward)
Learning Agent
New Vectorization Factors

Figure 3. The proposed framework for automatic vectorization with deep RL. The programs are read to extract the loops. The loop texts are fed to the code embedding generator to generate an embedding. The embedding is fed to the RL agent. The RL agent learns a policy that maps this embedding to optimal vectorization factors by injecting compiler pragmas and compiling the programs with Clang/LLVM to gather the rewards: the execution time improvements.

Table of contents

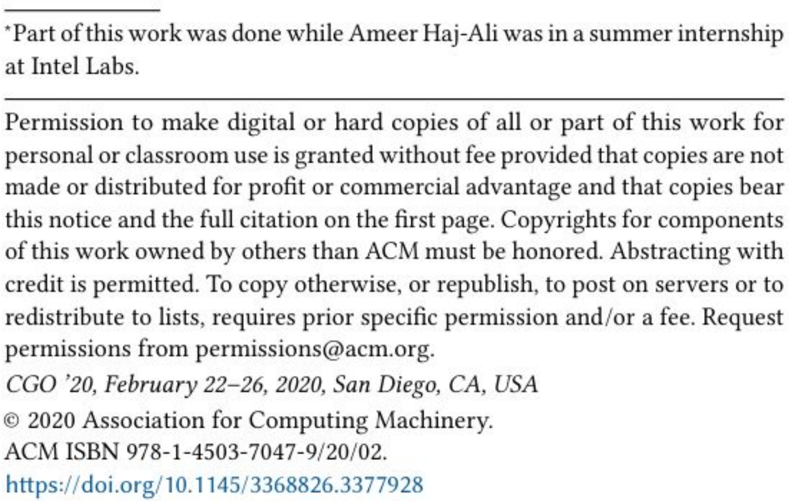
Abstract

1. Introduction
2. Motivation and Background
3. The Proposed Framework for Automatic Vectorization
4. Evaluation
5. Future Directions
6. Related work
7. Conclusion
8. Artifact description

My personal assessment by criteria

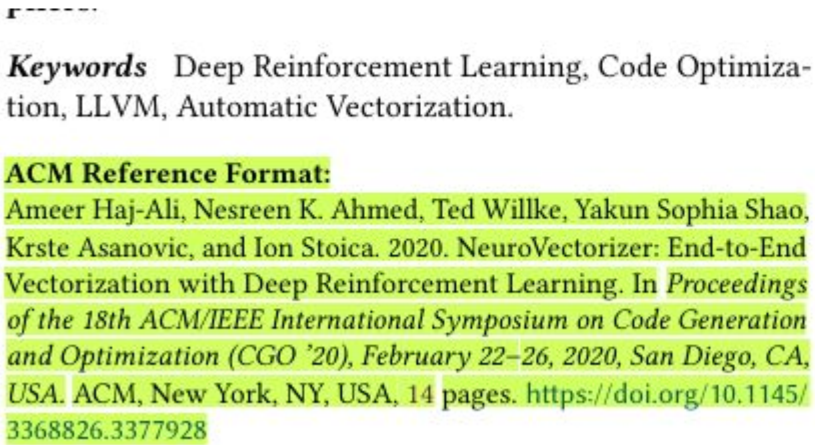
- Problem statement. 👍
- Innovation. 😐
- Contribution. 👍
- Logical correctness. 😐
- Proof of statements. 👍
- Readability. 👍

Features



[*] Part of this work was done while Ameer Haj-Ali was in a summer internship at Intel Labs.

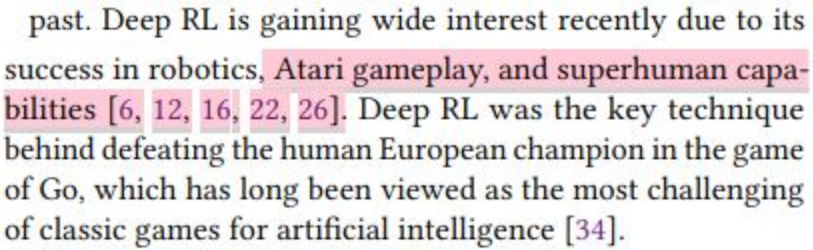
CGO '20, February 22–26, 2020, San Diego, CA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7047-9/20/02.
https://doi.org/10.1145/3368826.3377928



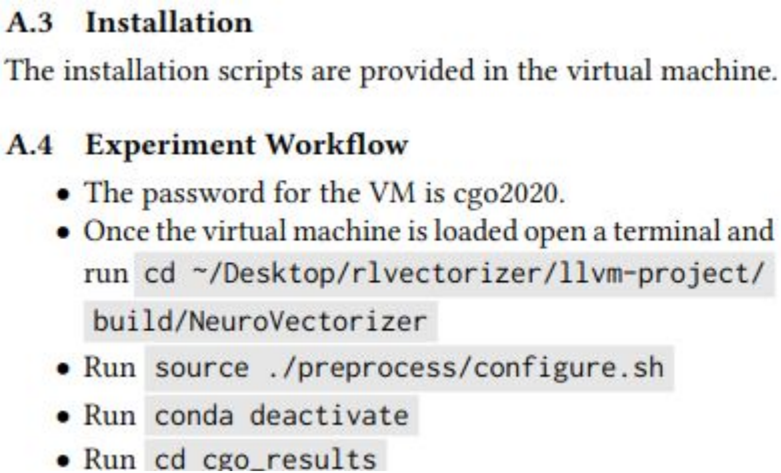
Keywords Deep Reinforcement Learning, Code Optimization, LLVM, Automatic Vectorization.

ACM Reference Format:
Ameer Haj-Ali, Nesreen K. Ahmed, Ted Willke, Yakun Sophia Shao, Krste Asanovic, and Ion Stoica. 2020. NeuroVectorizer: End-to-End Vectorization with Deep Reinforcement Learning. In *Proceedings of the 18th ACM/IEEE International Symposium on Code Generation and Optimization (CGO '20), February 22–26, 2020, San Diego, CA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3368826.3377928

Features



past. Deep RL is gaining wide interest recently due to its success in robotics, Atari gameplay, and superhuman capabilities [6, 12, 16, 22, 26]. Deep RL was the key technique behind defeating the human European champion in the game of Go, which has long been viewed as the most challenging of classic games for artificial intelligence [34].



A.3 Installation

The installation scripts are provided in the virtual machine.

A.4 Experiment Workflow

- The password for the VM is cgo2020.
- Once the virtual machine is loaded open a terminal and run `cd ~/Desktop/rlvectorizer/llvm-project/build/NeuroVectorizer`
- Run `source ./preprocess/configure.sh`
- Run `conda deactivate`
- Run `cd cgo_results`

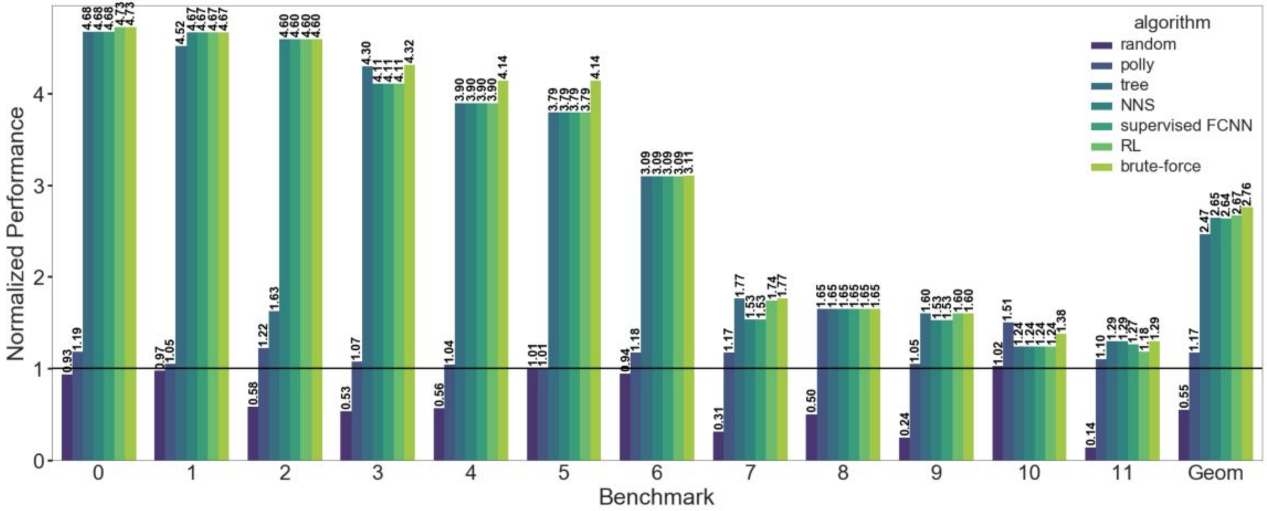


Figure 8. The performance of the proposed vectorizer that can be configured to use NNS, random search, decision trees, and RL compared to brute-force search, Polly and the baseline cost model. The performance is normalized to the baseline.

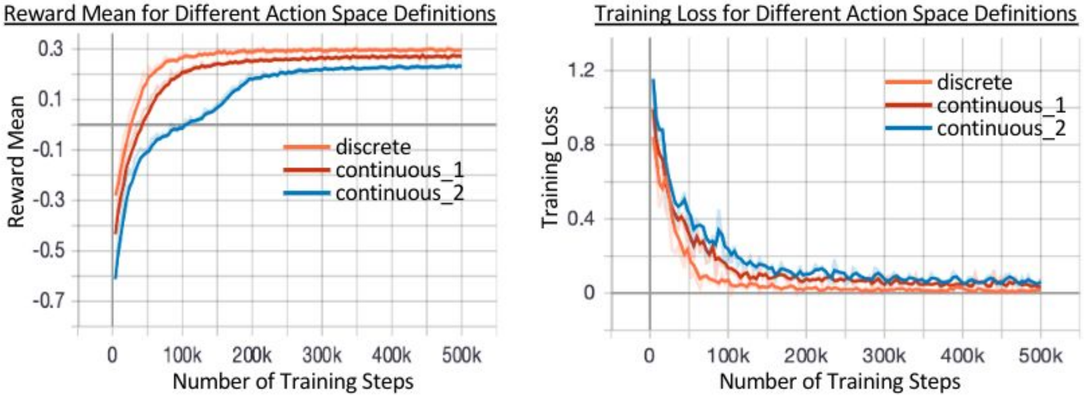


Figure 7. Reward mean and training loss for different action space definitions.