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

Paper by Ameer Haj-Ali, Nesreen K. Ahmed, Ted Willke, Yakun Sophia Shao, Krste Asanovic, Ion Stoica

Presentation by Reuben Gutmann, K. Faryab Haye, Ben Manley, Atreya Tata

Background

- Vector instructions: multiple basic operations simultaneously

```
r1 = ld(MEM[0])        ⟶    vr1 = ldV(MEM[0,4,8…])

r2 = add(r1, 1)        ⟶    vr2 = addV(vr1, 1)
```

- Loops: a common target for **“vectorization”**
  - **Vectorization Factor (VF):** How many instructions to pack together from different iterations
  - **Interleaving Factor (IF):** Stride of the memory accesses in the packed instructions

VF and IF

```
int32_t a[];
for (int i = 0; i < 100; ++i)
    a[i*3]++;
```

Interleaving Factor (IF) = 3

Unvectorized

```
temp0 = ld(a[0])
temp0 = add(temp0, 1)
stur(a[0], temp0)
...
temp99 = ld(a[99])
temp99 = add(temp99, 1)
stur(a[99], temp99)
```

100 iterations

Vectorized

```
temp0_4 = ldV(a[0,3,6,9])
temp0_4 = addV(temp0_4, 1)
sturV(a[0,3,6,9], temp0_4)
...
temp96_99 = ldV(a[96*3...99*3])
temp96_99 = addV(temp96_99, 1)
sturV(a[96*3...99*3], temp96_99)
```

25 iterations

Vectorization Factor (VF) = 4

Motivation

- Traditionally these hyperparameters are tuned using heuristics or expert hand tuning
- This leaves a ton of room for improvement

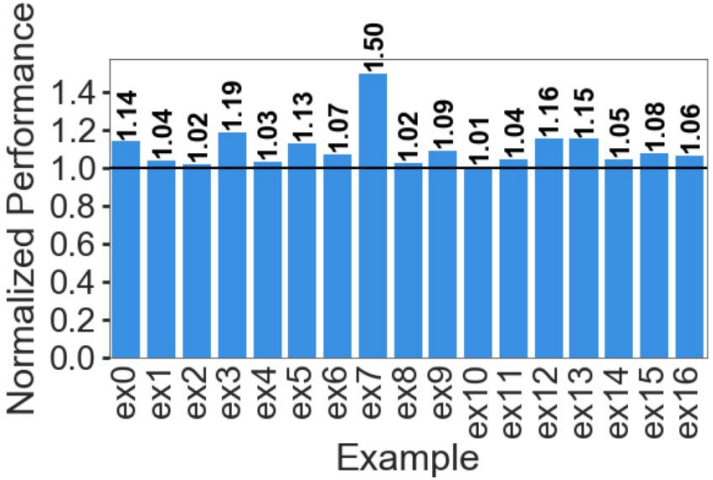


Figure 2. Performance of brute-force search of LLVM's vectorizer test suite, normalized to the baseline cost model implemented in LLVM.

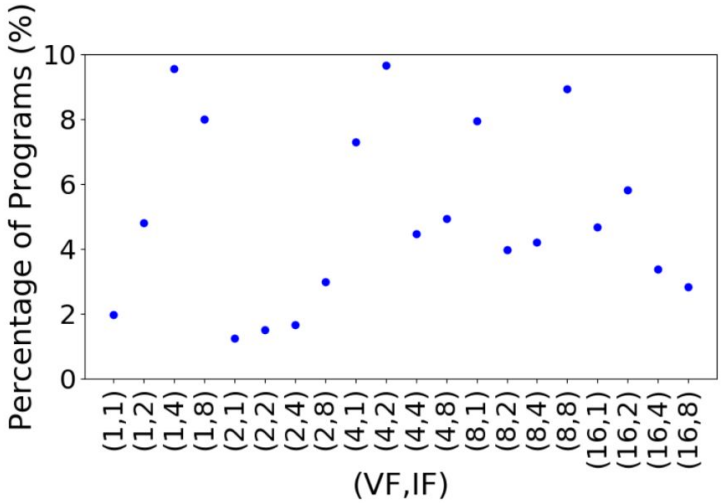
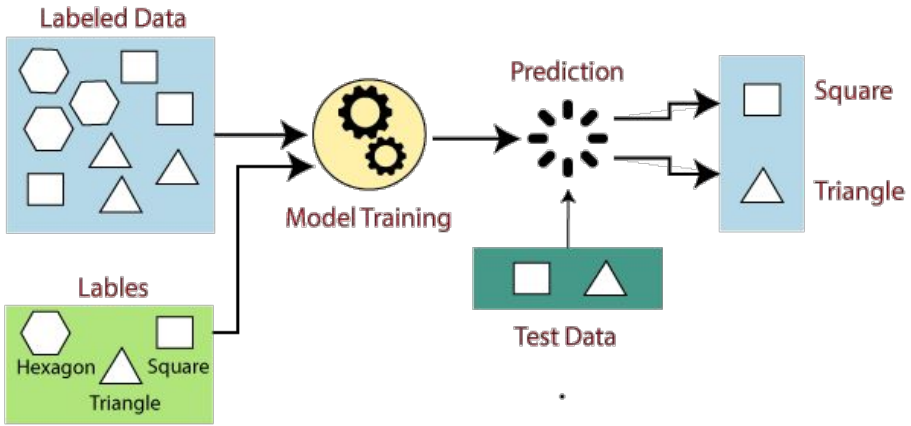


Figure 5. The distribution of optimal VF and IF with brute-force search for different programs in the dataset.

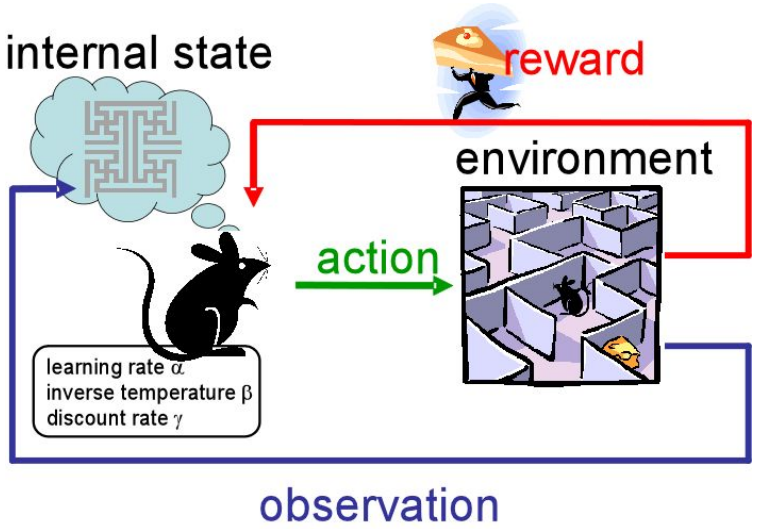
ML Refresher

Supervised Learning:



https://www.javatpoint.com/supervised-machine-learning

Reinforcement Learning:

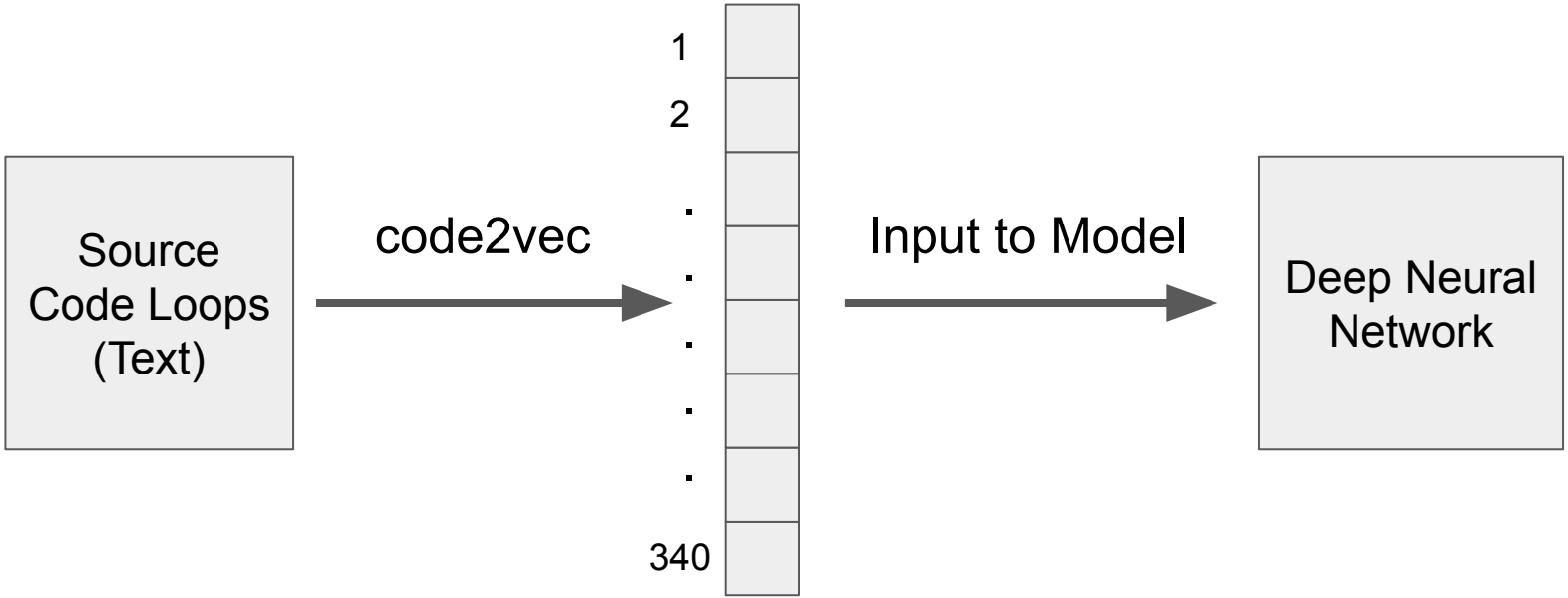


https://becominghuman.ai/the-very-basics-of-reinforcement-learning-154f28a79071?gi=5c84c0ee5db

Training Dataset

- NNs require a lot of training data!
- Synthetic dataset of more than 10,000 programs
  - Loops only!
- Used generators from LLVM Vectorization test suite
- Reduced noise in the code embeddings
  - Faster convergence

Code Embedding



code2vec is a pre-trained neural network

Method

State: Loop embedding (vector)
Action: (VF, IF) pair

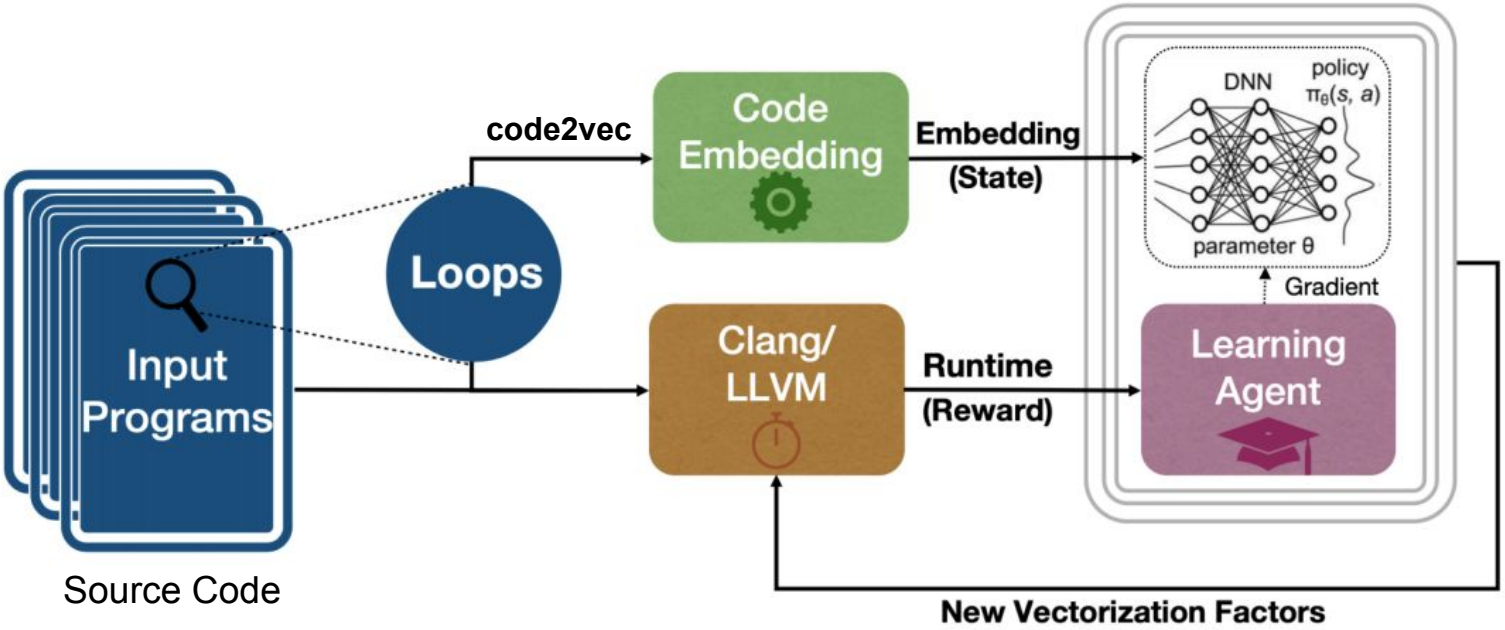


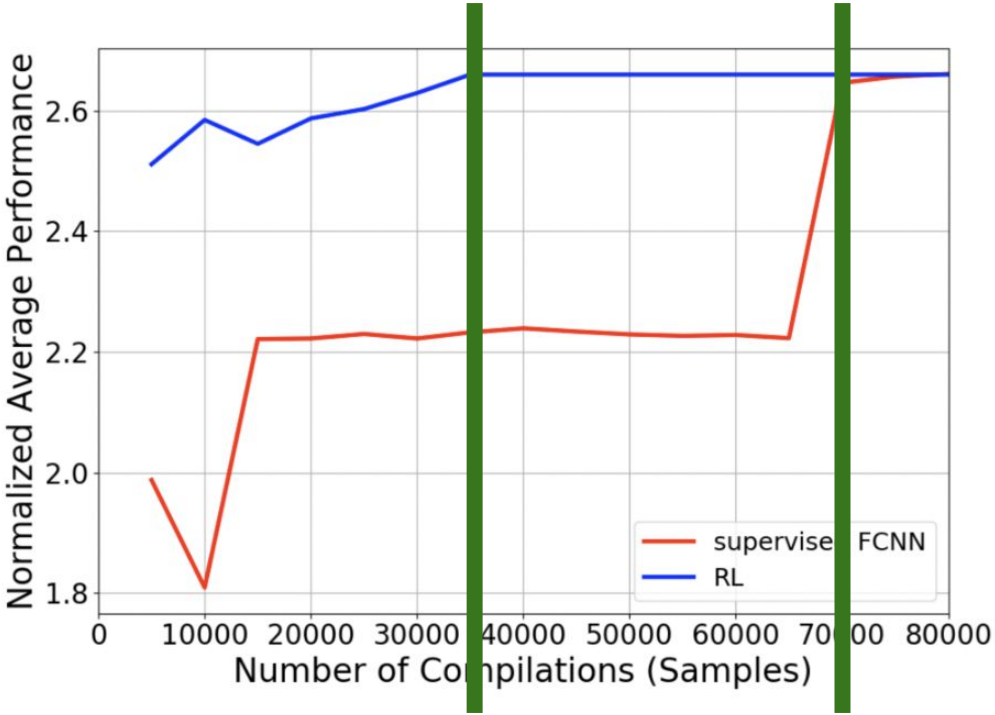
Figure 3 from the paper.

Method

```
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;
}
```

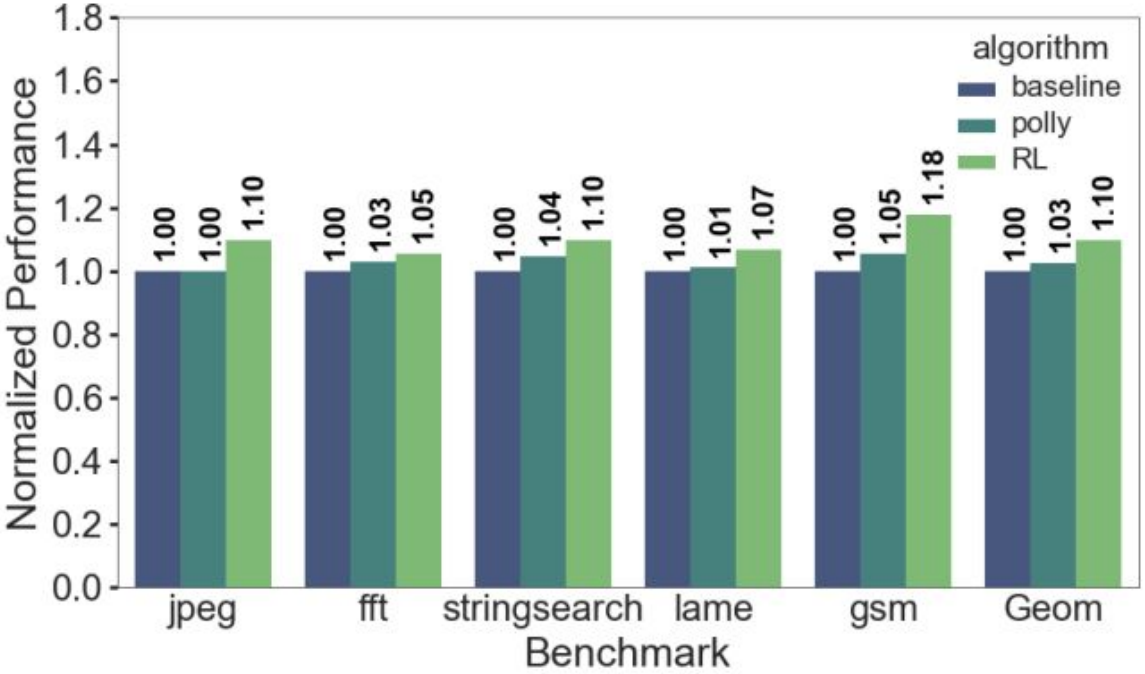
Results



- RL outperformed the baseline by **2.67×** on average
- Only **3%** worse than that of the brute-force search.
- Half the training time vs FCNN

Figure 9 from the paper.

Results



Method works well on never-before-seen benchmarks

Figure 11 from the paper.

Commentary

Strengths

- The evaluation was quite thorough.
  - They highlighted well where their RL model did not do better than other models.

- Good code reproducibility. They have a step by step checklist on their paper and a well documented repo.

Commentary

Weaknesses

- Doesn't address getting same accuracies in different models
- Code2vec embedding may not capture all important features about the loop

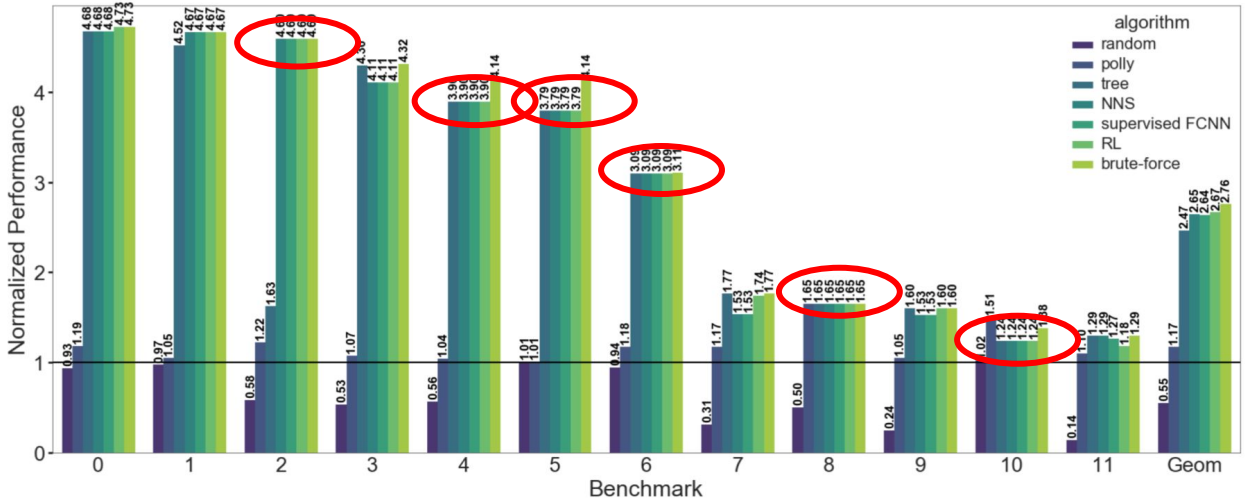


Figure 8. The performance of the proposed vectorizer that can be configured to use NNS, random search, decision trees, and RL compared to brute-force search, Polly and the baseline cost model. The performance is normalized to the baseline.

Q&A

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 22ś26, 2020, San Diego, CA, USA. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/ 3368826.3377928