AI for System and System for AI

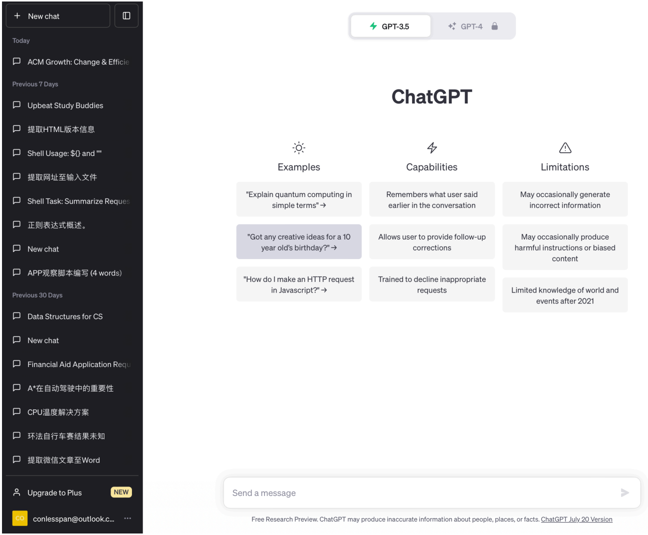
Conless Pan[†]

[†]ACM Class 2022
Shanghai Jiao Tong University

July 26, 2023

The Rise of Machine Learning

The Rise of Machine Learning



The Rise of Machine Learning

When we talk about the rise of machine learning, people usually raise these questions:

The Rise of Machine Learning

When we talk about the rise of machine learning, people usually raise these questions:

- What is machine learning?

The Rise of Machine Learning

When we talk about the rise of machine learning, people usually raise these questions:

- What is machine learning?
- Do you know its history?

The Rise of Machine Learning

When we talk about the rise of machine learning, people usually raise these questions:

- What is machine learning?
- Do you know its history?
- Why is it so important today?

The Rise of Machine Learning

When we talk about the rise of machine learning, people usually raise these questions:

- What is machine learning?
- Do you know its history?
- Why is it so important today?

But I don't want to talk about them, cause I'm not interested about AI.

Applications of Machine Learning

Anyway, AI is a useful tool.

Applications of Machine Learning

Anyway, AI is a useful tool.

- Generative AI

Applications of Machine Learning

Anyway, AI is a useful tool.

- Generative AI
- AI for science

Applications of Machine Learning

Anyway, AI is a useful tool.

- Generative AI
- AI for science
- Others

How can AI promote our research of computer system?

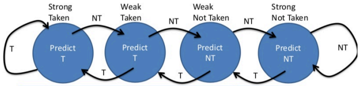
Let's compare these two games:[1]

[1] James E Smith. "A study of branch prediction strategies". In: *25 years of the international symposia on Computer architecture (selected papers)*. 1998, pp. 202–215.

How can AI promote our research of computer system?

Let's compare these two games:[1]

2-bit Saturating Counter



Iteration	Prediction	Actual	Mispredict?	State
1	NT	T	Y	Strong NT
2	NT	T	Y	Weak NT
3	T	T		Weak T
4	T	NT	Y	Strong T
...				
1	T	T		Weak T
2	T	T		Strong T
3	T	T		Strong T
4	T	NT	Y	Strong T

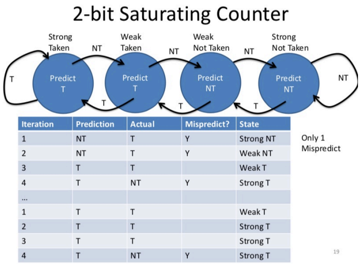
Only 1 Mispredict

19

[1] James E Smith. "A study of branch prediction strategies". In: *25 years of the international symposia on Computer architecture (selected papers)*. 1998, pp. 202–215.

How can AI promote our research of computer system?

Let's compare these two games:[1]



Iteration	Prediction	Actual	Mispredict?	State
1	NT	T	Y	Strong NT
2	NT	T	Y	Weak NT
3	T	T		Weak T
4	T	NT	Y	Strong T
...				
1	T	T		Weak T
2	T	T		Strong T
3	T	T		Strong T
4	T	NT	Y	Strong T

[1] James E Smith. "A study of branch prediction strategies". In: *25 years of the international symposia on Computer architecture (selected papers)*. 1998, pp. 202–215.

How can AI promote our research of computer system?

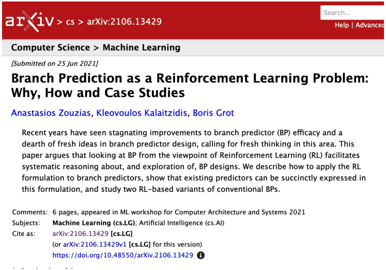
And it turns out that...[2][3]

[2] Anastasios Zouzias, Kleovoulos Kalaitzidis, and Boris Grot. "Branch Prediction as a Reinforcement Learning Problem: Why, How and Case Studies". In: *arXiv preprint arXiv:2106.13429* (2021).

[3] David Silver et al. "Mastering the game of go without human knowledge". In: *nature* 550.7676 (2017), pp. 354–359.

How can AI promote our research of computer system?

And it turns out that...[2][3]



arXiv > cs > arXiv:2106.13429

Search...

Help | Advanced

Computer Science > Machine Learning

[Submitted on 25 Jun 2021]

Branch Prediction as a Reinforcement Learning Problem: Why, How and Case Studies

Anastasios Zouzias, Kleovoulos Kalaitzidis, Boris Grot

Recent years have seen stagnating improvements to branch predictor (BP) efficacy and a dearth of fresh ideas in branch predictor design, calling for fresh thinking in this area. This paper argues that looking at BP from the viewpoint of Reinforcement Learning (RL) facilitates systematic reasoning about, and exploration of, BP designs. We describe how to apply the RL formulation to branch predictors, show that existing predictors can be succinctly expressed in this formulation, and study two RL-based variants of conventional BPs.

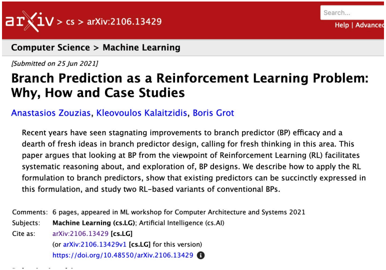
Comments: 6 pages, appeared in ML workshop for Computer Architecture and Systems 2021
Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI)
Cite as: arXiv:2106.13429 **[cs.LG]**
(or arXiv:2106.13429v1 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2106.13429

[2] Anastasios Zouzias, Kleovoulos Kalaitzidis, and Boris Grot. "Branch Prediction as a Reinforcement Learning Problem: Why, How and Case Studies". In: *arXiv preprint arXiv:2106.13429* (2021).

[3] David Silver et al. "Mastering the game of go without human knowledge". In: *nature* 550.7676 (2017), pp. 354–359.

How can AI promote our research of computer system?

And it turns out that...[2][3]


arXiv > cs > arXiv:2106.13429

Search...
Help | Advanced

Computer Science > Machine Learning

[Submitted on 25 Jun 2021]

Branch Prediction as a Reinforcement Learning Problem: Why, How and Case Studies

Anastasios Zouzias, Kleovoulos Kalaitzidis, Boris Grot

Recent years have seen stagnating improvements to branch predictor (BP) efficacy and a dearth of fresh ideas in branch predictor design, calling for fresh thinking in this area. This paper argues that looking at BP from the viewpoint of Reinforcement Learning (RL) facilitates systematic reasoning about, and exploration of, BP designs. We describe how to apply the RL formulation to branch predictors, show that existing predictors can be succinctly expressed in this formulation, and study two RL-based variants of conventional BPs.

Comments: 6 pages, appeared in ML workshop for Computer Architecture and Systems 2021
Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI)
Cite as: arXiv:2106.13429 **[cs.LG]**
(or arXiv:2106.13429v1 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2106.13429


nature

Explore content | About the journal | Publish with us | Subscribe

nature > articles > article

Published: 19 October 2017

Mastering the game of Go without human knowledge

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel & Demis Hassabis

Nature **550**, 354–359 (2017) | Cite this article

345k Accesses | **4205** Citations | **2528** Altmetric | Metrics

[2] Anastasios Zouzias, Kleovoulos Kalaitzidis, and Boris Grot. "Branch Prediction as a Reinforcement Learning Problem: Why, How and Case Studies". In: *arXiv preprint arXiv:2106.13429* (2021).

[3] David Silver et al. "Mastering the game of go without human knowledge". In: *nature* 550.7676 (2017), pp. 354–359.

How can AI promote our research of computer system?

Their similarities:

How can AI promote our research of computer system?

Their similarities:

1. Decision-making problems

How can AI promote our research of computer system?

Their similarities:

1. Decision-making problems
2. A finite state given as input

How can AI promote our research of computer system?

Their similarities:

1. Decision-making problems
2. A finite state given as input
3. A simple decision required as output

How can AI promote our research of computer system?

Their similarities:

1. Decision-making problems
2. A finite state given as input
3. A simple decision required as output

The role that AI plays: looking for a fitting function $y = f(x)$, which calculates the correct/best y with given x.

How does AI perform its job?

The perceptron was introduced at first:[4]

[4] Daniel A Jiménez and Calvin Lin. "Dynamic branch prediction with perceptrons". In: *Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture*. IEEE. 2001, pp. 197–206.

How does AI perform its job?

The perceptron was introduced at first:[4]

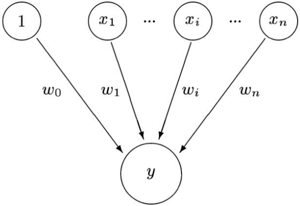


Figure 1: Perceptron Model. The input values $x_1, \ldots, x_n$, are propagated through the weighted connections by taking their respective products with the weights $w_1, \ldots, w_n$. These products are summed, along with the bias weight w_0, to produce the output value y.

[4] Daniel A Jiménez and Calvin Lin. "Dynamic branch prediction with perceptrons". In: *Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture*. IEEE. 2001, pp. 197–206.

How does AI perform its job?

The perceptron was introduced at first:[4]

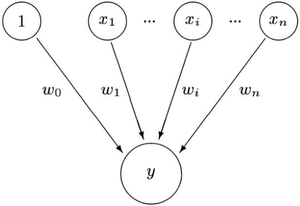
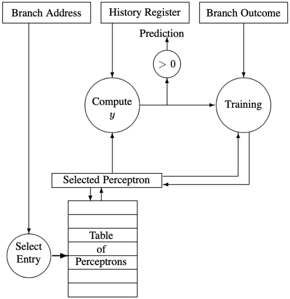


Figure 1: Perceptron Model. The input values $x_1, \ldots, x_n$, are propagated through the weighted connections by taking their respective products with the weights $w_1, \ldots, w_n$. These products are summed, along with the bias weight w_0, to produce the output value y.

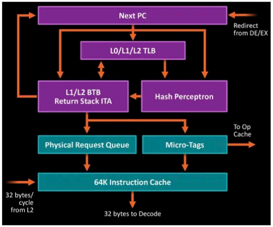


[4] Daniel A Jiménez and Calvin Lin. "Dynamic branch prediction with perceptrons". In: *Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture*. IEEE. 2001, pp. 197–206.

Applications

AMD Zen microarchitecture, the start of "AMD YES".

Front End [edit]



The Front End of the Zen core deals with the in-order operations such as instruction fetch and instruction decode. The instruction fetch is composed of two paths: a traditional decode path where instructions come from the instruction cache and a µOPs cache that are determined by the branch prediction (BP) unit. The instruction stream and the branch prediction unit track instructions in 64B windows. Zen is AMD's first design to feature a µOPs cache, a unit that not only improves performance, but also saves power (the µOPs cache was first introduced by Intel in their Sandy Bridge microarchitecture).

The branch prediction unit is decoupled and can start working as soon as it receives a desired operation such as a redirect, ahead of traditional instruction fetches. AMD still uses a hashed perceptron system similar to the one used in Jaguar and Bobcat, albeit likely much more finely tuned. AMD stated it's also larger than previous architectures but did not disclose actual sizes. Once the BP detects an indirect target operation, the branch is moved to the Indirect Target Array (ITA) which is 512 entry deep. The BP includes a 32-entry return stack.

In Zen, AMD moved the instruction TLB to BP (to much earlier in the pipeline than in previous architectures). This was done to allow for more-aggressive prefetching by allowing the physical address to be retrieved at an earlier stage. The BP is capable of storing 2 branches per BTB (Branch Target Buffer) entry, reducing the number of BTB reads necessary. ITLB is composed of:

Advanced version

Advanced neural network is also being introduced to many subfields of computer system...

Reinforcement is all you need

Let's go over the basic concept of reinforcement learning.

Reinforcement is all you need

Let's go over the basic concept of reinforcement learning.

- A virtual agent who makes decisions

Reinforcement is all you need

Let's go over the basic concept of reinforcement learning.

- A virtual agent who makes decisions
- A state space $S = \{s_i\}$

Reinforcement is all you need

Let's go over the basic concept of reinforcement learning.

- A virtual agent who makes decisions
- A state space $S = \{s_i\}$
- An action space $A = \{a_i\}$

Reinforcement is all you need

Let's go over the basic concept of reinforcement learning.

- A virtual agent who makes decisions
- A state space $S = \{s_i\}$
- An action space $A = \{a_i\}$
- Rewards $r_{a_i|s_i}$

Reinforcement is all you need

Let's go over the basic concept of reinforcement learning.

- A virtual agent who makes decisions
- A state space $S = \{s_i\}$
- An action space $A = \{a_i\}$
- Rewards $r_{a_i|s_i}$

When agent receives a reward or a feedback, it updates the estimation

$$\mathbb{E}(r_{a_i|s_i})$$

or under some situation the probability

$$\mathbb{P}(r_{a_i|s_i} > 0)$$

Applications

The problems that reinforcement learning can deal with:

- Heuristic
- Empirical

Database Management System

The configuration of the knobs[5]

	Category	Functionality	Example (Postgres)	Example (MySQL)
1	Access Control	Connections	max_connections	innodb_thread_concurrency
		Transactions	deadlock_timeout	innodb_table_locks
2	Query Optimizer	Query Plan	join_collapse_limit	rewriter_enabled
		Cost Values	seq_page_cost	join_buffer_size
3	Query Executor	Persistence	full_page_writes	replica_pending_jobs_size_max
4	Background Processes	Logging	log_rotation_size	binlog_cache_size
		Others	checkpoint_timeout	innodb_log_file_size
5	Resource (CPU)	CPU Usage	max_files_per_process	innodb thread concurrency
6	Resource (Memory)	Memory Space	shared_buffers	innodb_buffer_pool_size
7	Resource (Disk)	Disk IO/Caches	temp_file_limit	max_sort_file_size

RESEARCH-ARTICLE PUBLIC ACCESS

Automatic Database Management System Tuning Through Large-scale Machine Learning

Authors: Dana Van Aken, Andrew Pavlo, Geoffrey J. Gordon, Bohan Zhang Authors Info & Claims

SIGMOD '17: Proceedings of the 2017 ACM International Conference on Management of Data • May 2017 • Pages 1009–1024
• https://doi.org/10.1145/3035918.3064029

Published: 09 May 2017 Publication History

Check for updates

Operating System

The management of page table index[6]

Virtual Address Translation via Learned Page Table Indexes

Artemiy Margaritov† Dmitrii Ustiugov‡ Edouard Bugnion‡ Boris Grot†

†University of Edinburgh ‡EPFL

Abstract

Address translation is an established performance bottleneck [4] in workloads operating on large datasets due to frequent TLB misses and subsequent page table walks that often require multiple memory accesses to resolve. Inspired by recent research at Google on *Learned Index Structures* [14], we propose to accelerate address translation by introducing a new translation mechanism based on learned models using neural networks. We argue that existing software-based learned models are unable to outperform the traditional address translation mechanisms due to their high inference time, pointing toward the need for hardware-accelerated learned models. With a challenging goal to microarchitect a hardware-friendly learned page table index, we discuss a number of machine learning and systems trade-offs, and suggest future directions.

[6] Artemiy Margaritov et al. "Virtual address translation via learned page table indexes". In: *Conference on Neural Information Processing Systems*. 2018.

Applications

Anywhere we use heuristic to make a decision can be replaced by ml![7]

[7] Jeff Dean. "Machine learning for systems and systems for machine learning". In: *Presentation at 2017 Conference on Neural Information Processing Systems*. 2017.

Applications

Anywhere we use heuristic to make a decision can be replaced by ml![7]

- Compilers: instruction scheduling, register allocation, loop nest parallelization strategies, ...
- Networking: TCP window size decisions, backoff for retransmits, data compression, ...
- Operating systems: process scheduling, buffer cache insertion/replacement, file system prefetching, ...
- Job scheduling systems: which tasks/VMs to co-locate on same machine, which tasks to pre-empt, ...
- ASIC design: physical circuit layout, test case selection, ...

[7] Jeff Dean. "Machine learning for systems and systems for machine learning". In: *Presentation at 2017 Conference on Neural Information Processing Systems*. 2017.

Framework for RL in Sys

There are some frameworks for these systems[8]:

Park: An Open Platform for Learning-Augmented Computer Systems

Part of Advances in Neural Information Processing Systems 32 (NeurIPS 2019)

AuthorFeedback Bibtex MetaReview Metadata Paper Reviews Supplemental

Authors

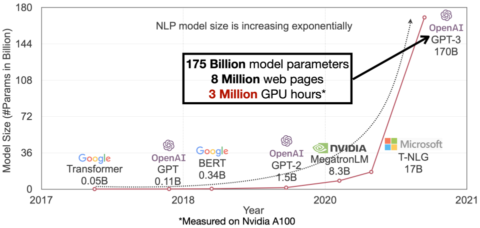
Hongzi Mao, Parimarjan Negi, Akshay Narayan, Hanrui Wang, Jiacheng Yang, Haonan Wang, Ryan Marcus, ravichandra addanki, Mehrdad Khani Shirkoohi, Songtao He, Vikram Nathan, Frank Cangialosi, Shaileshh Venkatakrishnan, Wei-Hung Weng, Song Han, Tim Kraska, Dr.Mohammad Alizadeh

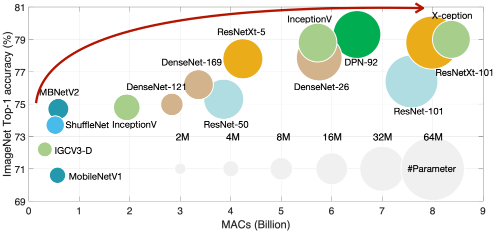
Abstract

We present Park, a platform for researchers to experiment with Reinforcement Learning (RL) for computer systems. Using RL for improving the performance of systems has a lot of potential, but is also in many ways very different from, for example, using RL for games. Thus, in this work we first discuss the unique challenges RL for systems has, and then propose Park an open extensible platform, which makes it easier for ML researchers to work on systems problems. Currently, Park consists of 12 real world system-centric optimization problems with one common easy to use interface. Finally, we present the performance of existing RL approaches over those 12 problems and outline potential areas of future work.

How can our works on system promote research of AI?

Large models are conquering ml...





Problems in system

With the growth of size, problems occurred.

Problems in system

With the growth of size, problems occurred.

- Low speed

Problems in system

With the growth of size, problems occurred.

- Low speed
- Limited memory

Problems in system

With the growth of size, problems occurred.

- Low speed
- Limited memory
- Popularization

Improvements

Speed can be achieved with loss of precision and generality

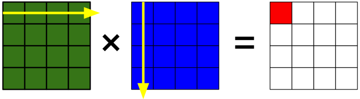
Reduced precision

about 1.2
× about 0.6
about 0.7

NOT

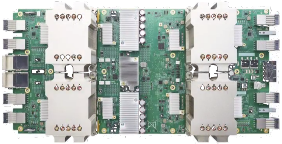
~~1.21042~~
~~× 0.61127~~
~~0.73989343~~

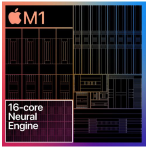
Specific operations



Improvements

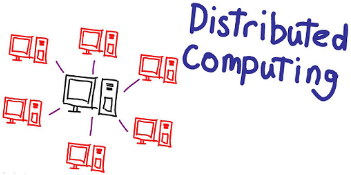
Thus we have Nvidia GPUs with CUDA units and tensor cores, Google TPU and Apple NPU.

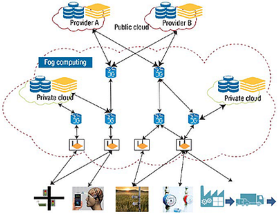


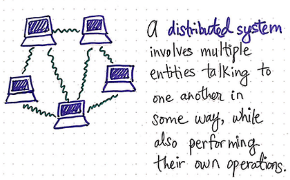


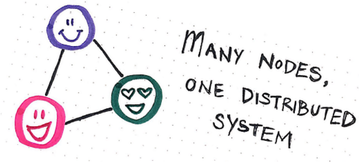
Improvements

In traditional data storage and computing, we distribute data/tasks to multiple machines for larger storage size and better speed.









Improvements

The same idea can also be applied into ml

Improvements

The same idea can also be applied into ml

- Distributed machine learning system (parallel compute / parallel data)

Improvements

The same idea can also be applied into ml

- Distributed machine learning system (parallel compute / parallel data)
- Computing device placement

Improvements

The same idea can also be applied into ml

- Distributed machine learning system (parallel compute / parallel data)
- Computing device placement
- Other topics in traditional distributed sys (communication, consistency)

Convenience

Programmers don't want to build their projects from every detail.

When writing cpp programs, it's convenient for us to use those frameworks and libraries.

AI programmers also need them.

Frameworks

NLTK

Spark

theano

Numpy

Keras

TensorFlow

Pytorch

Scikit-learn

Pandas

mxnet

NVIDIA

CUDA

tvm

Frameworks

cnn.py > ...

```python
from torch import nn
from torch.nn import functional as func


class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3, stride=1, padding=1)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
        self.fc1 = nn.Linear(64 * 7 * 7, 1024)
        self.fc2 = nn.Linear(1024, 512)
        self.fc3 = nn.Linear(512, 10)

    def forward(self, x):
        x = self.pool(func.relu(self.conv1(x)))
        x = self.pool(func.relu(self.conv2(x)))
        x = x.view(-1, 64 * 7 * 7)
        x = func.relu(self.fc1(x))
        x = func.relu(self.fc2(x))
        x = self.fc3(x)
        return x

net = CNN()
```

Higher level ideas

Microsoft

The Case for Learning-and-System Co-design

Chieh-Jan Mike Liang, Hui Xue, Mao Yang, Lidong Zhou

ACM SIGOPS Operating Systems Review | July 2019 , Vol 53(1): pp. 68-74

While decision-makings in systems are commonly solved with explicit rules and heuristics, machine learning (ML) and deep learning (DL) have been driving a paradigm shift in modern system design. Based on our decade of experience in operationalizing a large production cloud system, Web Search, learning fills the gap in comprehending and taming the system design and operation complexity. However, rather than just improving specific ML/DL algorithms or system features, we posit that the key to unlocking the full potential of learning-augmented systems is a principled methodology promoting learning-and-system co-design. On this basis, we present the AutoSys, a common framework for the development of learning-augmented systems.

Research

What is a good sys4ml/ml4sys research?

Research

What is a good sys4ml/ml4sys research?

- Should be both good AI and systems research

Research

What is a good sys4ml/ml4sys research?

- Should be both good AI and systems research
  - Provides insights to both communities

Research

What is a good sys4ml/ml4sys research?

- Should be both good AI and systems research
  - Provides insights to both communities
- Leverages understanding of both domains

Research

What is a good sys4ml/ml4sys research?

- Should be both good AI and systems research
  - Provides insights to both communities
- Leverages understanding of both domains
- I don't like adjust those parameters

Thank you!