Motivating the Next Paradigm

Opposing trends in past decade
- Machine learning boom
- Moore’s law bust

Machine learning supplants Moore’s law
- How do we close the loop?
ML Background
Literature review
  • How has machine learning been applied?
Analysis of current practice
  • What strategies are most effective?
Future work
  • Where do we go from here?
Agenda

ML Background

Literature review
- How has machine learning been applied?

Analysis of current practice
- What strategies are most effective?

Future work
- Where do we go from here?
ML Background

Fundamental applicability of ML
• Powerful, yet generic mathematical framework
• Task specification adaptation
  - IPC prediction example
ML Background

Diverse learning approaches & models

• Supervised
• Unsupervised
• Semi-supervised
• Reinforcement learning
ML Background: Supervised Learning

Diverse learning approaches & models

- Supervised
- Unsupervised
- Semi-supervised
- Reinforcement learning
Decision trees
- Tree structure
  - Node = feature
  - Branch = feature value(s)
- Simple, sequential, low overhead
ML Background: Supervised Learning

Bayesian networks

- Conditional relationships
  - Node = random variable
  - Edge = conditional dependence

- Scale w/ features
Support Vector Machines (SVMs)

- Optimize prediction margin
- Simplest case linear
  Use “kernel trick” for non-linear
ML Background: Supervised Learning

Neural networks
- Perceptron (one layer)
- Deep neural network (fully connected)
- Convolutional neural network (spatial aware convolution layers)
- Recurrent neural network (output → input loops)
ML Background: Unsupervised Learning

Diverse learning approaches & models

- Supervised
- Unsupervised
- Semi-supervised
- Reinforcement learning
ML Background: Semi-supervised Learning

Diverse learning approaches & models

- Supervised
- Unsupervised
- Semi-supervised
- Reinforcement learning
ML Background: Reinforcement Learning

Diverse learning approaches & models

- Supervised
- Unsupervised
- Semi-supervised
- Reinforcement learning
ML Background: Reinforcement Learning

Q-Learning
• Approximate optimal actions w/ stored action-value pairs
• Table-based

Deep Q-Learning
• Approximate optimal actions w/ neural network
• Weight storage
Agenda

ML Background

Literature review
  • How has machine learning been applied?

Analysis of current practice
  • What strategies are most effective?

Future work
  • Where do we go from here?
Application topics

- System simulation
- GPUs
- Memory systems & branch prediction
- Networks-on-Chip
- System-level optimization
- ML-enabled approximate computing
Literature Review: System Simulation

Reduced execution time vs cycle-accurate
  • Several orders of magnitude faster
  • Small accuracy penalty
    Ipek [8], Agarwal [86]

Mechanistic-empirical models
  • ML + other model types – easier & better
    Eyerman [10]

Cross-architecture predictions
  • Predict any architecture using available HW
    Zheng [11, 12]
Design space exploration
• Highly irregular scaling – no problem
  Wu [14], Jia [13], Jooya [15], Lin [16]

Cross-platform prediction (CPU → GPU)
• Save development time for important work
  Baldini [17], Ardalani [18] & [87]
Literature Review: GPUs

Scheduling
• High performance heterogeneous architecture
  Pattnaik [20]

Traffic pattern characterization
• Automatically identify prevalent patterns
  Li [90]
Caches (prefetch & re-use)
- Effective modeling for complex patterns
  Peled [21], Wang [24], Zeng & Guo [22], Teran [23], Braun & Litz [84], Bhatia [92]

Schedulers & control
- High-performance under constraints
  Ipek [25], Deng [29], Mukundan & Martinez [26], Ipek [27], Yoo [28], Yoo [30]

Branch prediction
- State-of-the-art accuracy
  St. Amant [31], Jimenez [32], Tarsa [85], Garza [93]
DVFS & link control
• Optimal proactive control
  Fettes [38], Savva [33], DiTomaso [34], Winkle [35], Rez [36], Clark [37]

Flow control
• Reduce latency and increase efficiency
  Daya [39], Yin [40]
Literature Review: Networks on Chip

Topology & general design
- Efficient exploration in vast design space
  Das [41], Joardar [43], Lin [44] & [91], Das [42], Rao [45]

Reliability
- Find optimal balance in diverse policies
  Wang [49] & [88], DiTomaso [48]
Literature Review: System Level Optimization

Energy efficiency optimization
• Significant energy reduction
• Minimal performance reduction

Won[50], Bai[55], Pan[51], Bailey[52], Lo[53], Mishra[54],
Chen & Marculescu[57], Chen[58], Imes[59], Tarsa [89]

Task allocation & resource management
• Consideration for long-term impact

Lu[60], Jain[65], Nemirovsky[61], Zhang[62], Bitirgen[63], Wang[64],
Ding [90]
Chip layout

• Replace standard design practice

Wu[66]
Literature Review: ML Enabled Approx Computing

Function approximation
• Reduce energy & execution time
• Small quality penalty
  Esmaeilzadeh [67], Yazdanbakhsh [68], Grigorian [69], Oliveira [71]

Statistical guarantees
• User controlled trade-offs, guaranteed quality
  Mahajan [70]
Agenda

ML Background

Literature review
  • How has machine learning been applied?

Analysis of current practice
  • What strategies are most effective?

Future work
  • Where do we go from here?
Analysis of Current Practice

Two general categories

• Online ML application (ML integrated)
  • Integrate ML at runtime
  • Limited by practical constraints

• Offline ML application (ML supported)
  • Support architecture during design
  • Generally higher complexity
Current Practice: Online Applications

Model selection
• Primarily supervised or RL
• Some tasks can use either (w/ limitations) [38]
• Some tasks supervised only
Current Practice: Online Applications

Implementation & overhead

• Dedicated vs opportunistic data collection
  [39], [50]

• Hardware vs software
  [67], [50]

• Hardware trade-offs
  [33], [31], [25], [26]
Current Practice: Online Applications

Optimization

- Mitigate online learning side effects
  - Update model, not system
    [21]
  - Initial alternate controller
    [50]
Model/feature selection
• Not limited by hardware constraints → substantial diversity
• Design space exploration
  - Iterative search
    [41], [42], [43], [44], [91]
  - Standard prediction
    [15], [45], [47]
• Some tasks supervised only
Optimization

• Improve data efficiency & model accuracy
• Ensembles
  - Subset choices & outlier removal
    [15], [18]
• Sampling (avoid systematic bias)
  [47]

Current Practice: Offline Applications
Mechanistic-empirical
• Simple & avoid assumptions
• High accuracy
  [10], [45]

Task specific considerations
• Complex feature handling
  [29]
• Result interpretation
  [18]
ML Background

Literature review
  • How has machine learning been applied?

Analysis of current practice
  • What strategies are most effective?

Future work
  • Where do we go from here?
Future Work: Implementation

New strategies $\rightarrow$ effective application

- Pruning
  - Train complex models
  - Implement simple models
Future Work: Improvements

New models & architecture-aware techniques
- Hierarchical models
  - Model high & low level details
- Phase-level & nanosecond scale
  - Finer-grain prediction/control
Future Work: Tools

Tools

- Ideas limited by application complexity
- General purpose framework → More accessible implementation
Future Work: Applications

Applications

• Extend existing approaches
  - Emerging technologies & architectures
  - System-level approximate computing

• Long-term potential
  - System-wide co-optimization
  - Automated design
Conclusion

Broad applicability

Many opportunities

Future potential - automated architecture
References Cited


References Cited


References Cited


References Cited


Additional References


Additional References


Additional References


AI for Architecture:
Principles and Prospects for the Next Paradigm

Drew Penney and Lizhong Chen
College of Engineering        System Technology and Architecture Research (STAR) Lab
Oregon State University
Motivation
• Route at source using loops
  How to configure these loops?
• Evolutionary is unreliable
• Heuristics are inflexible
Why deep reinforcement learning?
- No training set
- Effective/flexible exploration framework
Implementation

- State/action/reward representation
  - State: N*N NoC → $N^2 \times N^2$ hop count matrix
  - Action: 2 opposing points = rectangle
  - Reward: Loops =
    good (unless constraint violated)
    low hop count = good
Case Study: ML-Enabled Routerless NoC Design

Results

- 4x4 NoC = seconds
- 10x10 NoC = minutes
- Highly regular & high diversity
- 3.2x higher throughput, 1.6x lower latency, 5x lower power compared to mesh

Lin [91]