Boosting Automated Reasoning using Machine Learning

Martin Suda

Czech Technical University in Prague, Czech Republic

CIIRC, March 2021

A Wake-up Picture

Artificial Intelligence
$\rightarrow$ **Automated Reasoning**

Gottfried Leibniz's dream: *Calculus ratiocinator*

Artificial Intelligence
$\rightarrow$ **Automated Reasoning**

Gottfried Leibniz's dream: *Calculus ratiocinator*

$\rightarrow$ **Automated Deduction**

Symbolic, Logical Calculi, "Sound and Complete", Undecidable!, . . .

Artificial Intelligence
$\rightarrow$ **Automated Reasoning**

Gottfried Leibniz's dream: *Calculus ratiocinator*

$\rightarrow$ **Automated Deduction**

Symbolic, Logical Calculi, "Sound and Complete", Undecidable!, ...

$\rightarrow$ **Automated Theorem Proving (ATP)**

(as opposed to, e.g., the Interactive Theorem Proving)

Artificial Intelligence
$\rightarrow$ **Automated Reasoning**

Gottfried Leibniz's dream: *Calculus ratiocinator*

$\rightarrow$ **Automated Deduction**

Symbolic, Logical Calculi, "Sound and Complete", Undecidable!, ...

$\rightarrow$ **Automated Theorem Proving (ATP)**

(as opposed to, e.g., the Interactive Theorem Proving)

$\rightarrow$ **for First-order Logic**

(there is also HO, there are non-classical, modal, temporal, ...)

Artificial Intelligence
$\rightarrow$ **Automated Reasoning**

Gottfried Leibniz's dream: *Calculus ratiocinator*

$\rightarrow$ **Automated Deduction**

Symbolic, Logical Calculi, "Sound and Complete", Undecidable!, ...

$\rightarrow$ **Automated Theorem Proving (ATP)**

(as opposed to, e.g., the Interactive Theorem Proving)

$\rightarrow$ **for First-order Logic**

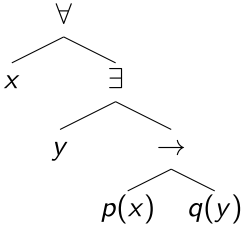
(there is also HO, there are non-classical, modal, temporal, ...)

$\rightarrow$ **Saturation-based ATPs (for FO logic)**

The Flavours of ML in Automated Theorem Proving

Our basic data structures are (primarily) logical formulas

- symbolic expression: "$\forall x \exists y . p(x) \rightarrow q(y)$"
- in fact, a tree-like object:

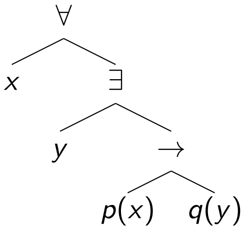


- drawn from an infinite (enumerable) universe

The Flavours of ML in Automated Theorem Proving

Our basic data structures are (primarily) logical formulas

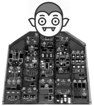
- symbolic expression: "$\forall x \exists y . p(x) \rightarrow q(y)$"
- in fact, a tree-like object:



- drawn from an infinite (enumerable) universe

How to apply modern ML to this?

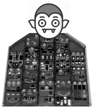
- hand-crafted features
- recursive neural networks
- graph convolutional networks
- ...



Vampire

- Automatic Theorem Prover (ATP) for First-order Logic (FOL) with equality and theories
- state-of-the-art saturation-based prover

Deepire: Powering ATPs using Neural Networks

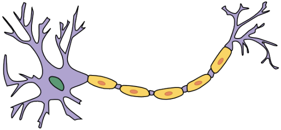


Vampire

- Automatic Theorem Prover (ATP) for First-order Logic (FOL) with equality and theories
- state-of-the-art saturation-based prover

Neural (internal) guidance

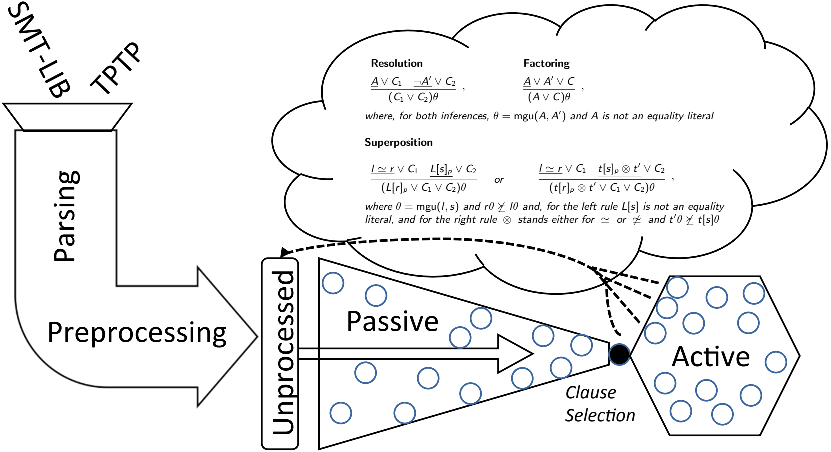
- targeting the clause selection decision point
- supervised learning from successful prover runs



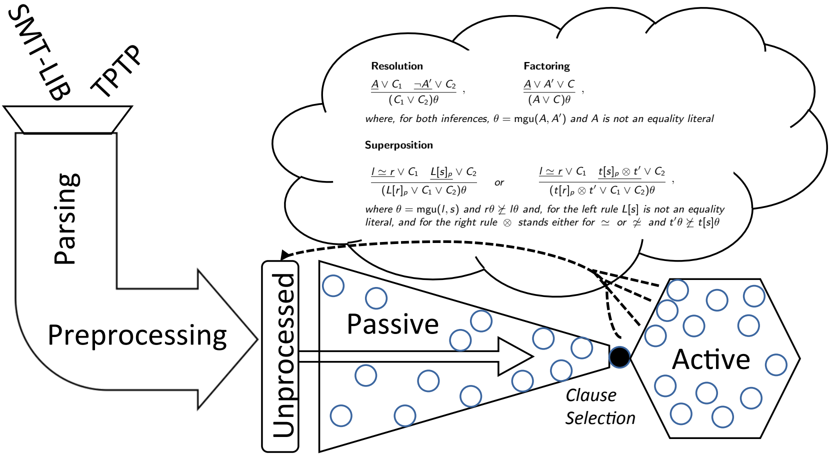
Outline

Outline

Saturation-based Theorem Proving in One Slide



Saturation-based Theorem Proving in One Slide



At a typical successful end: $|Passive| \gg |Active| \gg |Proof|$

Clause selection: traditionally and neurally

Traditionally: simple clause evaluation criteria

- weight: prefer clauses with fewer symbols
- age: prefer clauses that were generated long time ago
- ...

Combine these using priority queues into a single scheme

Clause selection: traditionally and neurally

Traditionally: simple clause evaluation criteria

- weight: prefer clauses with fewer symbols
- age: prefer clauses that were generated long time ago
- ...

Combine these using priority queues into a single scheme

How to improve this with ML?

- train a classifier for recognizing clauses that appeared in past proofs (as opposed to those selected, but not found useful)
- integrate into the selection mechanism,
  prioritizing clauses classified positively

Outline

Started off with ENIGMA:

- ENIGMA: Efficient Learning-Based Inference Guiding Machine [Jakubův&Urban,2017]
- ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E [Chvalovský et al.,2019]
- ENIGMA Anonymous: Symbol-Independent Inference Guiding Machine [Jakubův et al.,2020]

See also:

- Deep Network Guided Proof Search [Loos et al.,2017]
- Property Invariant Embedding for Automated Reasoning [Olšák et al.,2020]

Machine Learning Guided Clause Selection

Started off with ENIGMA:

- ENIGMA: Efficient Learning-Based Inference Guiding Machine [Jakubův&Urban,2017]
- ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E [Chvalovský et al.,2019]
- ENIGMA Anonymous: Symbol-Independent Inference Guiding Machine [Jakubův et al.,2020]

See also:

- Deep Network Guided Proof Search [Loos et al.,2017]
- Property Invariant Embedding for Automated Reasoning [Olšák et al.,2020]

Most recently also Deepire:

- New Techniques that Improve ENIGMA-style Clause Selection Guidance (submitted to CADE)
- Vampire With a Brain Is a Good ITP Hammer (submitted to ITP)

Similar to ENIGMA:

- build a classifier for recognising good clauses
- good are those that appeared in past proofs

Deepire: Neurally Guided Clause Selection in Vampire

Similar to ENIGMA:

- build a classifier for recognising good clauses
- good are those that appeared in past proofs

Deepire's contributions:

- Learn from clause derivation trees only
  Not looking at what it says, just who its ancestors were.

Deepire: Neurally Guided Clause Selection in Vampire

Similar to ENIGMA:

- build a classifier for recognising good clauses
- good are those that appeared in past proofs

Deepire's contributions:

- Learn from clause derivation trees only
  Not looking at what it says, just who its ancestors were.
- Integrate using layered clause queues
  A smooth improvement of the base clause selection strategy.
- Tree Neural Networks: constant work per derived clause
- A signature agnostic approach
- Lazy evaluation trick (not all derived need to be evaluated)

Similar to ENIGMA:

- build a classifier for recognising good clauses
- good are those that appeared in past proofs

Deepire's contributions:

- Learn from clause derivation trees only
  Not looking at what it says, just who its ancestors were.
- Integrate using layered clause queues
  A smooth improvement of the base clause selection strategy.
- Tree Neural Networks: constant work per derived clause
- A signature agnostic approach
- Lazy evaluation trick (not all derived need to be evaluated)

Preliminary Evaluation on Mizar "57880"

- Learn from 63595 proofs of 23071 problems (three 30s runs)
- Deepire solves 26217 (i.e. +4054) problems in a single 10s run

Outline

Some High-level Remarks

Things to consider:

- Evaluation speed
- Can the choices depend on proof state?
- How exactly is the new advice integrated into the ATP?

Some High-level Remarks

Things to consider:

- Evaluation speed
- Can the choices depend on proof state?
- How exactly is the new advice integrated into the ATP?

"**Looping**" typically helps quite a bit!

- basically (a form of) Reinforcement Learning already

Some High-level Remarks

Things to consider:

- Evaluation speed
- Can the choices depend on proof state?
- How exactly is the new advice integrated into the ATP?

"Looping" typically helps quite a bit!

- basically (a form of) Reinforcement Learning already

A question to the audience, maybe?

- a classifier (yes/no) seems sub-optimal here
- Are there any good architectures for "unbounded" regressors over recursively defined inputs?

Some High-level Remarks

Things to consider:

- Evaluation speed
- Can the choices depend on proof state?
- How exactly is the new advice integrated into the ATP?

"Looping" typically helps quite a bit!

- basically (a form of) Reinforcement Learning already

A question to the audience, maybe?

- a classifier (yes/no) seems sub-optimal here
- Are there any good architectures for "unbounded" regressors over recursively defined inputs?

(Similarly: a fixed number of rounds of message passing in a GCN for an arbitrary formula also does not "feel right".)

Even Higher-level

Relation to AGI?

Logic as a Means to Explainable AI?

Embeddings Respecting Semantic Logical Relations?

One More Picture

One More Picture

Thank you for attention!