Deep Reasoning
A Vision for Automated Deduction
Stephan Schulz
Deep Reasoning

A Vision for Automated Deduction

Wer Visionen hat, sollte zum Arzt gehen!
Deep Reasoning

A Vision for Automated Deduction

Anybody with visions should go see a doctor!
Agenda

- Introduction
- Deep Learning
- Automated Theorem Proving
- Deep Reasoning
- Conclusion
Introduction: Historical Perspective

1955 Logic Theorist
1956 Dartmouth Workshop - “Birth of AI”
1957 Perceptron
1958 LISP
1960 Davis-Putnam (DPLL 1962)
1965 Resolution/Unification
1970 Knuth-Bendix Completion
1972 PROLOG (1983 WAM)
1965-1975 MLP/back propagation
1980s Expert systems/Planners
1986 Decision tree learning
1990-1994 Superposition calculus
since 1997 Development of E (E 0.3 January 1999)
since ca. 2005 “Deep Learning”
2008 E 1.0
Deep Learning
Deep Learning - Introduction

- Instance of machine learning
- Typical setting: Supervised learning
  - Large number of pre-classified examples
  - Examples are presented with expected output
  - System learns classification/evaluation
- Result: Trained model
  - Will provide classification/evaluation when presented with new input
Deep Learning - Methods

- Application of known techniques on a new scale
  - Supervised learning (classification/evaluation/association)
  - Artificial neural networks
  - Gradient-based learning/back-propagation
- New:
  - Big networks
  - Complex network structure
    - Multiple sub-networks
    - Convolution layers
    - Recurrence
  - (Mostly) raw input
    - Feature extraction is part of the learning
    - Encoding is part of the learning
Deep Learning - Successes

- AI used to have problems with “easy” tasks
- Deep learning successfully addresses these problems
  - Image recognition
  - Voice recognition
  - Natural language translation
  - Hard games
    - Video games (real time)
    - Go
    - Poker
Deep Learning - Successes

- AI used to have problems with “easy” tasks
- Deep learning successfully addresses these problems
  - Image recognition
  - Voice recognition
  - Natural language translation
  - Hard games
    - Video games (real time)
    - Go
    - Poker

Deep learning drives resurgence of Artificial Intelligence!
Deep Learning - Why Now?

- Popularity of Deep Learning
  - ... slowly growing since the mid 2000s
  - ... explosively growing since mid 2010s

- Driven by “big hardware”
  - Clusters of computers
  - ... with clusters of GPUs

- Driven by “big data”
  - Large training sets
  - Large size of individuals

- Driven by Open Source
  - Algorithms and models published under permissive licenses
  - Many state-of-the-art machine learning libraries available
Deep Learning - A Parable

Cast of Characters
Deep Learning - A Parable

Cast of Characters

Neanderthal Man
Deep Learning - A Parable

Cast of Characters

Neanderthal Man

Sir Isaac Newton
Deep Learning - A Parable

Cast of Characters

Neanderthal Man

Sir Isaac Newton

Dr. Albert Einstein
Neanderthal Learning
Neanderthal Learning
Neanderthal Learning
Neanderthal Learning
Neanderthal Learning
Neanderthal Learning

Don’t sit under tree!
Ugh!
Neanderthal Learning

Don't sit under tree!

Round things fall down!

Ugh!
Enlightenment!
Enlightenment!
Enlightenment!
Enlightenment!
Enlightenment!
Enlightenment!

\[ F = ma \]

\[ F = G \frac{m_1 m_2}{r^2} \]
Compare and Contrast
Compare and Contrast
Compare and Contrast

\[ F = ma \]

\[ F = G \frac{m_1 m_2}{r^2} \]
Compare and Contrast

\[ E = mc^2 \]

\[ G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu} \]
Compare and Contrast
Compare and Contrast

Round things fall down!
Ugh!
What an interesting early human. I wonder what he thinks!
Deep Learning Weaknesses

- Computationally expensive
  - Big models use specialized hardware for training
  - Even model application has non-trivial cost
- Knowledge is represented by large set distributed weights
  - Low inherent level of abstraction
  - Model is noisy
- Knowledge is largely inaccessible
  - Hard to understand
  - Hard to explain
  - Hard to communicate
Deep Learning Weaknesses

- Computationally expensive
  - Big models use specialized hardware for training
  - Even model application has non-trivial cost
- Knowledge is represented by large set distributed weights
  - Low inherent level of abstraction
  - Model is noisy
- Knowledge is largely inaccessible
  - Hard to understand
  - Hard to explain
  - Hard to communicate

Unsupported claim (still true):
Deep learning alone will run into natural limits!
Theorem Proving: Big Picture

Real World Problem

8 X : human ( X ) ! mortal ( X )
8 X : philosopher ( X ) ! human ( X )
philosopher ( socrates )
? | = mortal ( socrates )

Real World Problem
Formalized Problem

ATP
Proof Search
Proof
Countermodel
Timeout
or
or
Real World Problem
Theorem Proving: Big Picture

Real World Problem

Formalized Problem
Real World Problem

Formalized Problem

\( \forall X : human(X) \rightarrow mortal(X) \)
\( \forall X : philosopher(X) \rightarrow human(X) \)
\( \text{philosopher}(\text{socrates}) \)
\( \text{?} \)
\( \models \)
\( \text{mortal}(\text{socrates}) \)
Theorem Proving: Big Picture

Real World Problem

Formalized Problem

\[ \forall X : \text{human}(X) \rightarrow \text{mortal}(X) \]
\[ \forall X : \text{philosopher}(X) \rightarrow \text{human}(X) \]
\[ \text{philosopher}(\text{socrates}) \]
\[ \Rightarrow \]
\[ \text{mortal}(\text{socrates}) \]
Theorem Proving: Big Picture

Real World Problem

Formalized Problem

∀X : human(X) → mortal(X)
∀X : philosopher(X) → human(X)
philosopher(socrates)
?
|= mortal(socrates)
Theorem Proving: Big Picture

Real World Problem

Formalized Problem

\[ \forall X : human(X) \rightarrow mortal(X) \]
\[ \forall X : philosopher(X) \rightarrow human(X) \]
philosopher(socrates)

? \models mortal(socrates)
Theorem Proving: Big Picture

Real World Problem

Formalized Problem

\[ \forall X : \text{human}(X) \rightarrow \text{mortal}(X) \]
\[ \forall X : \text{philosopher}(X) \rightarrow \text{human}(X) \]
\[ \text{philosopher}(\text{socrates}) \]
\[ ? \]
\[ \models \]
\[ \text{mortal}(\text{socrates}) \]

Proof or Countermodel

ATP
Theorem Proving: Big Picture

**Real World Problem**

∀X : human(X) → mortal(X)
∀X : philosopher(X) → human(X)
philosopher(socrates)

? |=
mortal(socrates)

**Formalized Problem**

∀X : human(X) → mortal(X)
∀X : philosopher(X) → human(X)
philosopher(socrates)

| |
--- |
? |=
mortal(socrates)

**Proof**

or

**Countermodel**

or

**Timeout**

| ATP | 15 |
Logics of Interest

- Propositional logic
  - SAT-solving: relatively independent sub-field

- First-order logics
  - ... with free symbols
  - ... with free symbols and equality
  - ... with background theories
  - ... with free symbols and background theories

- Higher order logics
  - Currently developing field
Contradiction and Saturation

- **Proof by contradiction**
  - Assume negation of conjecture
  - Show that axioms and negated conjecture imply falsity

- **Saturation**
  - Convert problem to Clause Normal Form
  - Systematically enumerate logical consequences of axioms and negated conjecture
  - Goal: Explicit contradiction (empty clause)

- **Redundancy elimination**
  - Use contracting inferences to simplify or eliminate some clauses
Contradiction and Saturation

- Proof by contradiction
  - Assume negation of conjecture
  - Show that axioms and negated conjecture imply falsity

- Saturation
  - Convert problem to Clause Normal Form
  - Systematically enumerate logical consequences of axioms and negated conjecture
  - Goal: Explicit contradiction (empty clause)

- Redundancy elimination
  - Use contracting inferences to simplify or eliminate some clauses

Search control problem: How and in which order do we enumerate consequences?
Proof Search

# SZS output start CNFRefutation
fof(pel55_4, axiom, (![X1]:![X2]:((killed(X1,X2)=>hates(X1,X2))));
   file(’PUZ001+1.p’, pel55_4)).
...
fof(pel55, conjecture, (killed(agatha,agatha)),
   file(’PUZ001+1.p’, pel55)).
...
fof(c_0_12, plain, ((lives(esk1_0)&killed(esk1_0,agatha))),
   inference(skolemize,[status(esa)],
   [inference(variable_rename,[status(thm)],[pel55_1]))]).
...
cnf(c_0_14,plain,(hates(X1,X2)|~killed(X1,X2)),
   inference(split_conjunct,[status(thm)],[c_0_11]))).
...
cnf(c_0_23,plain,(hates(esk1_0,agatha)),
   inference(spm,[status(thm)],[c_0_14, c_0_15]))).
...
cnf(c_0_45,plain,($false),
   inference(sr,[status(thm)],[inference(rw,[status(thm)],
      [c_0_15, c_0_43]), c_0_44]), [’proof’]).
# SZS output end CNFRefutation
First-order logic is semi-decidable

- Provers search for proof in infinite space
- ... of possible derivations
- ... of possible consequences

Major choice points of Superposition calculus:

- Term ordering (which terms are bigger)
- (Negative) literal selection
- Selection of clauses for inferences (with the given clause algorithm)
Some Properties of ATP

- Individual operations cheap(ish)
  - Computing one consequence is no problem
  - Computing 1000 consequences is no problem
- But: Large/infinite search space
  - 1000 consequences is usually enough for a proof
  - ...but rarely enough to find it!
- Combinatorial explosion
  - High branching factor
  - Simplification helps a lot
  - ...but not nearly enough!
Big Data and ATP

- Automated tuning of theorem provers since the 1990s
  - Examples:
    - E-SETHEO schedules
    - E’s automatic auto mode
    - Vampire’s *black magic* box
  - Based on performance only
- Reason: Proof search traces are big!
  - ...really big!
  - ...and theorem provers are memory-limited anyways
Big Data and ATP

- Automated tuning of theorem provers since the 1990s
  - Examples:
    - E-SETHEO schedules
    - E’s automatic auto mode
    - Vampire’s black magic box
  - Based on performance only
- Reason: Proof search traces are big!
  - ... really big!
  - ... and theorem provers are memory-limited anyways
- Ca. 2014: Something wonderful happens
  - Hardware finally catches up
  - Implementation techniques improve

What is wrong? The prover is not running out of memory!
Big Data and ATP

- Automated tuning of theorem provers since the 1990s
  - Examples:
    - E-SETHEO schedules
    - E’s automatic auto mode
    - Vampire’s *black magic* box
  - Based on performance only
- Reason: Proof search traces are big!
  - ... really big!
  - ... and theorem provers are memory-limited anyways
- Ca. 2014: Something wonderful happens
  - Hardware finally catches up
  - Implementation techniques improve

*We can finally afford to look DEEPLY into proofs!*
Deep Reasoning
Vision: Search Control

▶ Long-term goal: Extract search control knowledge
  ▶ ...from examples of successful proof searches
  ▶ ...from examples of failing proof searches

▶ Primary use case: Clause selection
  ▶ Which of the current candidate consequences should be considered first?
  ▶ Extract good/bad search decisions from proof protocols
Vision: Search Control

- Long-term goal: Extract search control knowledge
  - ...from examples of successful proof searches
  - ...from examples of failing proof searches

- Primary use case: Clause selection
  - Which of the current candidate consequences should be considered first?
  - Extract good/bad search decisions from proof protocols

- It’s happening!
  - Premise selection (Urban, Irving, et al)
  - Clause Selection (Loos, Irvin, Kaliszyk et al) - see next session
Vision: Search Control

- Long-term goal: Extract search control knowledge
  - ... from examples of successful proof searches
  - ... from examples of failing proof searches
- Primary use case: Clause selection
  - Which of the current candidate consequences should be considered first?
  - Extract good/bad search decisions from proof protocols
- It’s happening!
  - Premise selection (Urban, Irving, et al)
  - Clause Selection (Loos, Irvin, Kaliszyk et al) - see next session
Vision: Automated Scientist

- Setting: Background theory + examples
  - Background theory in explicit logic
  - Examples

- Process
  - Deep learner hypothesizes relationship
  - Hypothesis is converted to symbolic logic (*Magic happens here*)
  - ATP system checks hypotheses for consistency with background theory
    - Failure: Abduction can refine hypothesis
    - Success: Tentatively add hypothesis to theory
  - ATP system generates new consequences to test on examples
Vision: Fully Interactive AI

- **Setting**: Rational agent interacting with environment
- **Deep learner**:
  - Vision
  - Voice
  - Language
  - Suggest actions
- **Symbolic reasoning system**
  - Hard-coded world knowledge
  - Hard-coded constraints on behavior
The End
Conclusion

- Deep learning and symbolic reasoning are complementary
- Hardware is now finally sufficient for both
  - ...even in combined systems
- We’re looking forward to an interesting future
Deep learning and symbolic reasoning are complementary

Hardware is now finally sufficient for both
  - . . . even in combined systems

We’re looking forward to an interesting future

And when the time comes to decide whether to switch on the new, improved AI that is vastly superior to humans and will eliminate all errors, a couple of imperial bureaucrats will gather round a table, and one will say: “We’ve already paid for it, so let’s switch it on” . . .
Conclusion

- Deep learning and symbolic reasoning are complementary
- Hardware is now finally sufficient for both
  - . . . even in combined systems
- We’re looking forward to an interesting future

And when the time comes to decide whether to switch on the new, improved AI that is vastly superior to humans and will eliminate all errors, a couple of imperial bureaucrats will gather round a table, and one will say: “We’ve already paid for it, so let’s switch it on” . . .

Marc Uwe Kling (as “the Kangaroo”)
Thank you!
Questions? Discussion?