Automated Theorem Proving:

A Deep Learning Approach

Industrial Sponsor: Real AI

Gwang Hyeon CHOI, Seung Yong MOON, Zheng PAN, Justin SUN

fppt.com

Outline

- Introduction
- Holophrasm: The Program
  - Metamath Database
  - Proof Tree
  - The Neural Networks
- Improving Holophrasm: TFLearn
- Conclusion

fppt.com

Outline

- Introduction
- Holophrasm: The Program
  - Metamath Database
  - Proof Tree
  - The Neural Networks
- Improving Holophrasm: TFLearn
- Conclusion

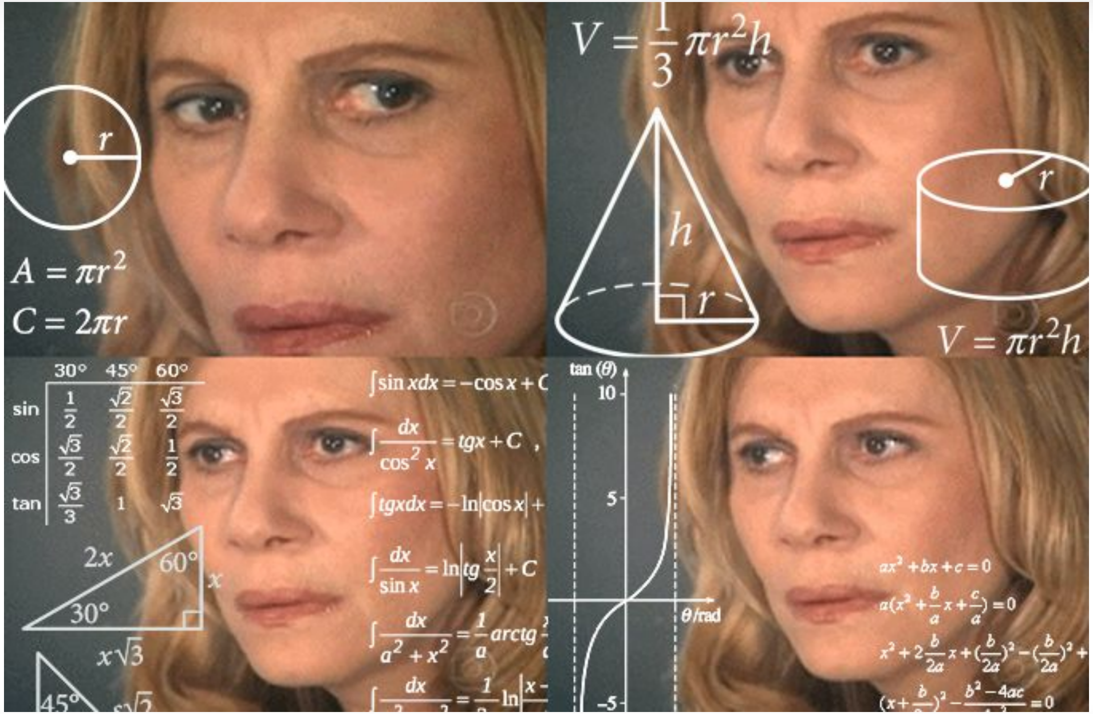
fppt.com

Company Background

- Aims to develop safe and beneficial Artificial General Intelligence (AGI)
- Monitors deep learning development
- Conducts research, like Automated Theorem Proving

fppt.com

Proving Math Theorems ...



fppt.com

Inspiration: AlphaGo

Monte Carlo Tree Search + Artificial Neural Networks

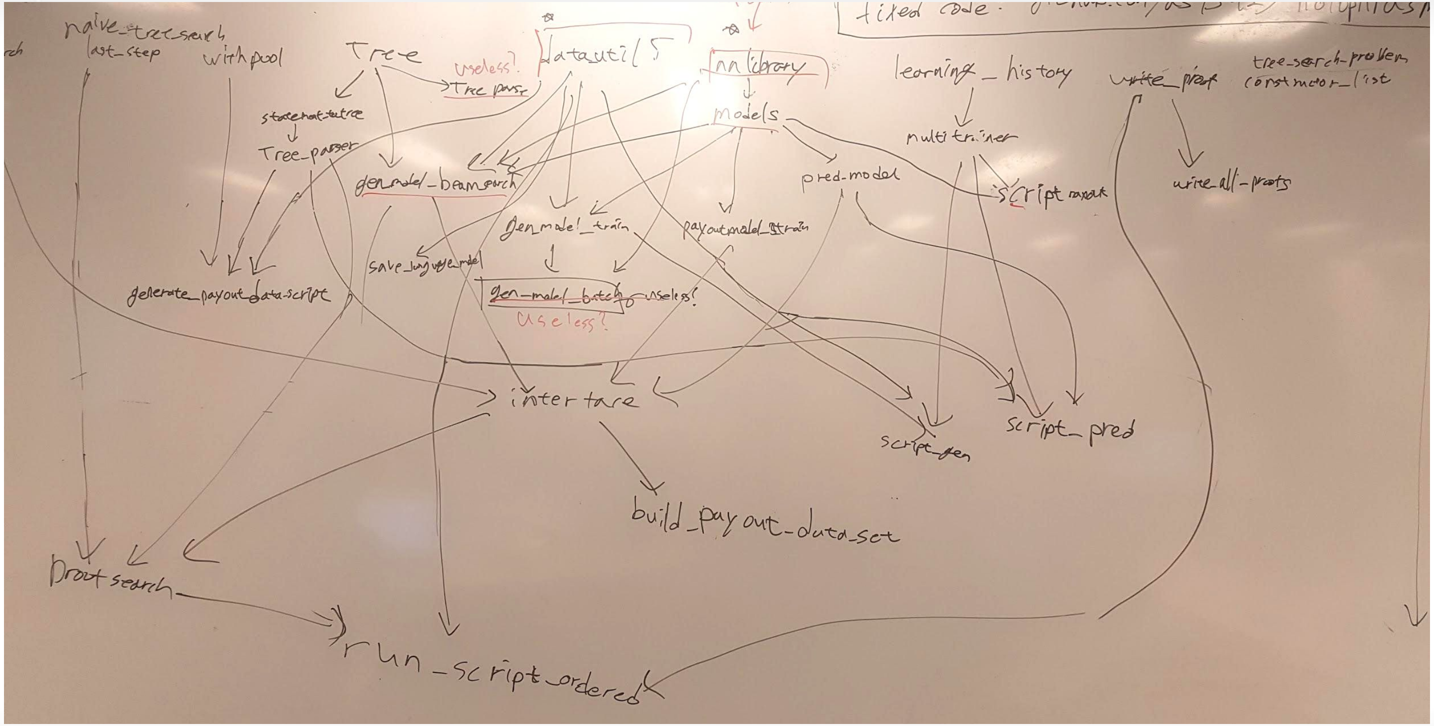
Image Source: https://www.theguardian.com/technology/2017/may/23/alphago-google-ai-beats-ke-jie-china-go

Holophrasm

- Written in Python by Daniel P. Z. Whalen, in 2016
- Tree search
- Three neural networks
- Finds proofs for 14.3% of the test propositions

fppt.com

Holophrasm



fppt.com

Objectives

- Study the codes and understand how it works

- Try to improve the program

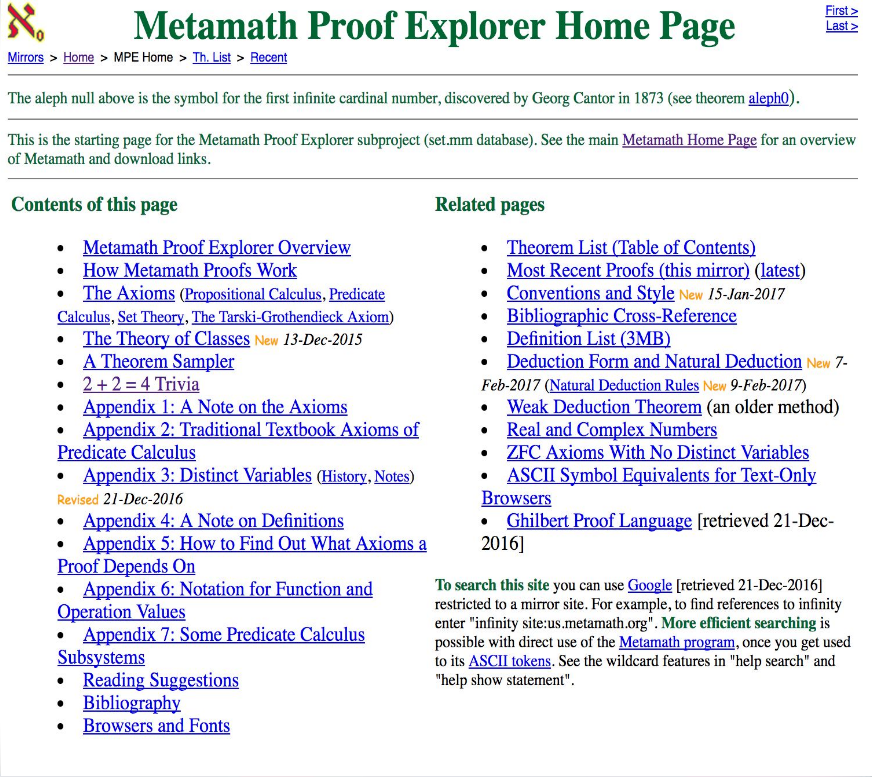
Outline

- Introduction
- Holophrasm: The Program
  - Metamath Database
  - Proof Tree
  - The Neural Networks
- Improving Holophrasm: TFLearn
- Conclusion

fppt.com

Metamath Database

- Over 19,000 propositions
- Built from 22 axioms



First >
Last >

ℵ0 **Metamath Proof Explorer Home Page**

Mirrors > Home > MPE Home > Th. List > Recent

The aleph null above is the symbol for the first infinite cardinal number, discovered by Georg Cantor in 1873 (see theorem aleph0).

This is the starting page for the Metamath Proof Explorer subproject (set.mm database). See the main Metamath Home Page for an overview of Metamath and download links.

Contents of this page

- Metamath Proof Explorer Overview
- How Metamath Proofs Work
- The Axioms (Propositional Calculus, Predicate Calculus, Set Theory, The Tarski-Grothendieck Axiom)
- The Theory of Classes New *13-Dec-2015*
- A Theorem Sampler
- 2 + 2 = 4 Trivia
- Appendix 1: A Note on the Axioms
- Appendix 2: Traditional Textbook Axioms of Predicate Calculus
- Appendix 3: Distinct Variables (History, Notes) Revised *21-Dec-2016*
- Appendix 4: A Note on Definitions
- Appendix 5: How to Find Out What Axioms a Proof Depends On
- Appendix 6: Notation for Function and Operation Values
- Appendix 7: Some Predicate Calculus Subsystems
- Reading Suggestions
- Bibliography
- Browsers and Fonts

Related pages

- Theorem List (Table of Contents)
- Most Recent Proofs (this mirror) (latest)
- Conventions and Style New *15-Jan-2017*
- Bibliographic Cross-Reference
- Definition List (3MB)
- Deduction Form and Natural Deduction New *7-Feb-2017* (Natural Deduction Rules New *9-Feb-2017*)
- Weak Deduction Theorem (an older method)
- Real and Complex Numbers
- ZFC Axioms With No Distinct Variables
- ASCII Symbol Equivalents for Text-Only Browsers
- Ghilbert Proof Language [retrieved 21-Dec-2016]

To search this site you can use Google [retrieved 21-Dec-2016] restricted to a mirror site. For example, to find references to infinity enter "infinity site:us.metamath.org". **More efficient searching** is possible with direct use of the Metamath program, once you get used to its ASCII tokens. See the wildcard features in "help search" and "help show statement".

fppt.com

Example Proposition

- Proposition (context)
  - Hypotheses
  - Assertion
  - Proof

Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \rightarrow \psi)$
mp2b.3	$\vdash (\psi \rightarrow \chi)$

Assertion

Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \rightarrow \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \rightarrow \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

http://us.metamath.org/mpegif/mp2b.html

fppt.com

Tautological Proposition

- Proposition (context)

  ~~Hypotheses~~

  Assertion

  Proof

Theorem 2p2e4 9811

Assertion

Ref	Expression
2p2e4	$\vdash (2 + 2) = 4$

Proof of Theorem 2p2e4

Step	Hyp	Ref	Expression
1		df-2 9773	..3 $\vdash 2 = (1 + 1)$
2	1	oveq2i 5804	.2 $\vdash (2 + 2) = (2 + (1 + 1))$
3		df-4 9775	..3 $\vdash 4 = (3 + 1)$
4		df-3 9774	...4 $\vdash 3 = (2 + 1)$
5	4	oveq1i 5803	..3 $\vdash (3 + 1) = ((2 + 1) + 1)$
6		2cn 9785	...4 $\vdash 2 \in \mathbb{C}$
7		ax-1cn 8764	...4 $\vdash 1 \in \mathbb{C}$
8	6, 7, 7	addassi 8814	..3 $\vdash ((2 + 1) + 1) = (2 + (1 + 1))$
9	3, 5, 8	3eqtri 2282	.2 $\vdash 4 = (2 + (1 + 1))$
10	2, 9	eqtr4i 2281	1 $\vdash (2 + 2) = 4$

fppt.com

Axiomatic Proposition

- Proposition (context)

  Hypotheses

  Assertion

  ~~Proof~~

Axiom ax-mp 10

Hypotheses

Ref	Expression
min	$\vdash \varphi$
maj	$\vdash (\varphi \rightarrow \psi)$

Assertion

Ref	Expression
ax-mp	$\vdash \psi$

http://us.metamath.org/mpegif/ax-mp.html

fppt.com

Proof Step

Type I:

A context hypothesis

- No theorem is applied

Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \rightarrow \psi)$
mp2b.3	$\vdash (\psi \rightarrow \chi)$

Assertion

Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \rightarrow \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \rightarrow \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

http://us.metamath.org/mpegif/mp2b.html

fppt.com

Proof Step

Type II:

An assertion a

- A theorem is applied
- Usually entailed by previous step(s)

Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \to \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

http://us.metamath.org/mpegif/mp2b.html

fppt.com

Substitutions

Notations:

C : Context a : A Type II step in C	T : Theorem used to derive a a_T : Assertion of T e_T : Hypotheses of T

fppt.com

Substitutions

ϕ: A set of substitutions satisfying $\phi(a_T) = a$.

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem **mp2b**

Step	Hyp	Ref	Expression
1		mp2b.1	3 $\vdash \varphi$
2		mp2b.2	3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	2 $\vdash \psi$
4		mp2b.3	2 $\vdash (\psi \to \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

a

Context C

Axiom **ax-mp** 10

Hypotheses

Ref	Expression
min	$\vdash \varphi$
maj	$\vdash (\varphi \to \psi)$

Assertion

Ref	Expression
ax-mp	$\vdash \psi$

a_T

Detailed syntax breakdown of Axiom **ax-mp**

Step	Hyp	Ref	Expression
1		wps	1 wff ψ

Theorem T

fppt.com

Substitutions

- Constrained variables (*cv*):
  - appear in a_T
- Unconstrained variables (*uv*):
  - not appear in a_T

Hypotheses

Ref	Expression
mp2b.1	⊢ φ
mp2b.2	⊢ $(\varphi \rightarrow \psi)$
mp2b.3	⊢ $(\psi \rightarrow \chi)$

Assertion

Ref	Expression
mp2b	⊢ χ

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	3 ⊢ φ
2		mp2b.2	3 ⊢ $(\varphi \rightarrow \psi)$
3	1, 2	ax-mp 10	2 ⊢ ψ
4		mp2b.3	2 ⊢ $(\psi \rightarrow \chi)$
5	3, 4	ax-mp 10	1 ⊢ χ

Context C

Axiom ax-mp 10

Hypotheses

Ref	Expression
min	⊢ φ
maj	⊢ $(\varphi \rightarrow \psi)$

Assertion

Ref	Expression
ax-mp	⊢ ψ

Detailed syntax breakdown of Axiom ax-mp

Step	Hyp	Ref	Expression
1		wps	1 wff ψ

Theorem T

- Constrained substitutions:
  - replaces a *cv*
- Unconstrained substitutions:
  - replaces a *uv*

fppt.com

Outline

- Introduction
- Holophrasm: The Program
  - Metamath Database
  - Proof Tree
  - The Neural Networks
- Improving Holophrasm: TFLearn
- Conclusion

fppt.com

Proof Tree

Every Metamath proof has a tree structure.

fppt.com

Constructing a Proof Tree

Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \to \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

Constructing a Proof Tree

$\vdash \chi$

Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \to \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

fppt.com

Constructing a Proof Tree

$\vdash \chi$

Axiom ax-mp 10

Hypotheses

Ref	Expression
min	$\vdash \varphi$
maj	$\vdash (\varphi \to \psi)$

Assertion

Ref	Expression
ax-mp	$\vdash \psi$

Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \to \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

fppt.com

Constructing a Proof Tree

$\vdash \chi$

Substitutions

Replace ψ by χ
Replace φ by ψ

Axiom ax-mp 10

Hypotheses

Ref	Expression
min	$\vdash \varphi$
maj	$\vdash (\varphi \to \psi)$

Assertion

Ref	Expression
ax-mp	$\vdash \psi$

Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

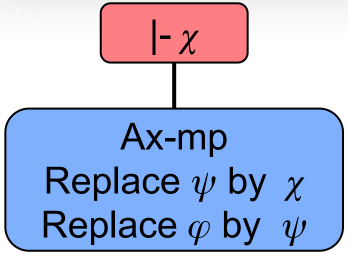
Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \to \chi)$
5	3, 4	$\boxed{\text{ax-mp 10}}$	1 $\vdash \chi$

fppt.com

Constructing a Proof Tree



Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

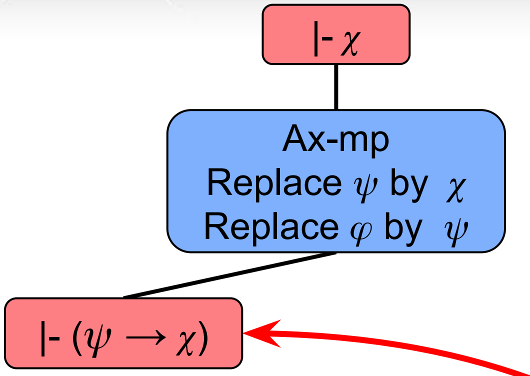
Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \to \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

fppt.com

Constructing a Proof Tree



Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

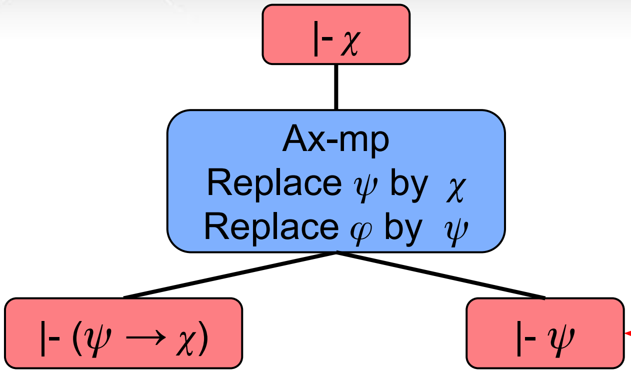
Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \to \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

fppt.com

Constructing a Proof Tree



Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \rightarrow \psi)$
mp2b.3	$\vdash (\psi \rightarrow \chi)$

Assertion

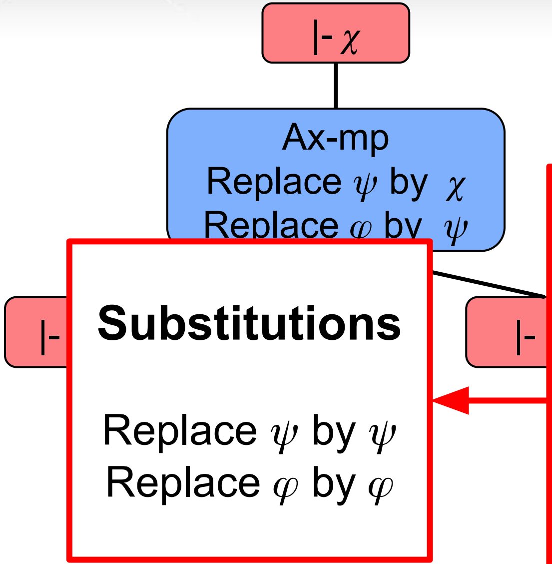
Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \rightarrow \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \rightarrow \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

fppt.com

Constructing a Proof Tree



Axiom ax-mp 10

Hypotheses

Ref	Expression
min	$\vdash \varphi$
maj	$\vdash (\varphi \rightarrow \psi)$

Assertion

Ref	Expression
ax-mp	$\vdash \psi$

Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \rightarrow \psi)$
mp2b.3	$\vdash (\psi \rightarrow \chi)$

Assertion

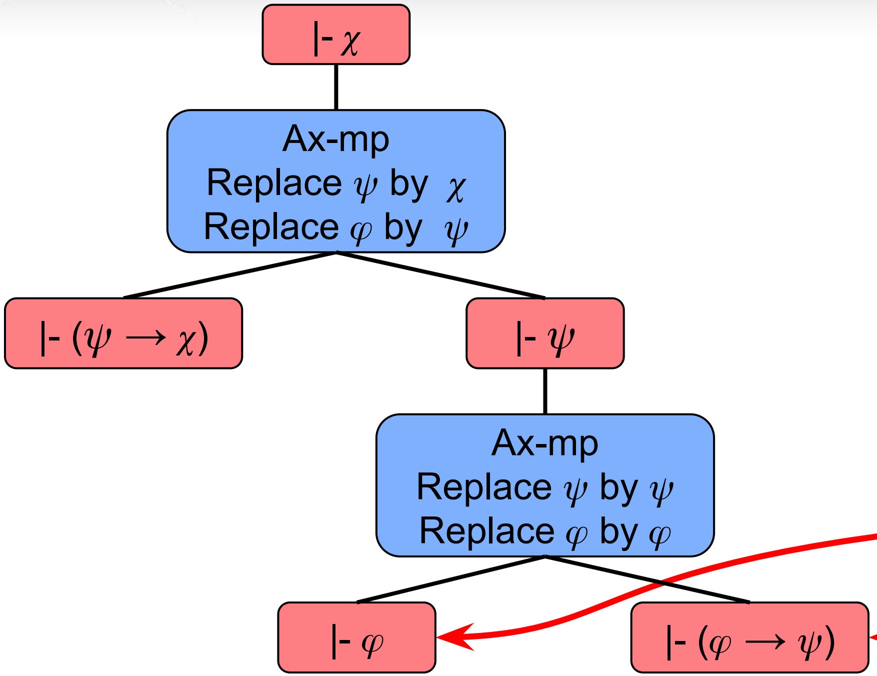
Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \rightarrow \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \rightarrow \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

fppt.com

Constructing a Proof Tree



Theorem mp2b 11

Hypotheses

Ref	Expression
mp2b.1	$\vdash \varphi$
mp2b.2	$\vdash (\varphi \to \psi)$
mp2b.3	$\vdash (\psi \to \chi)$

Assertion

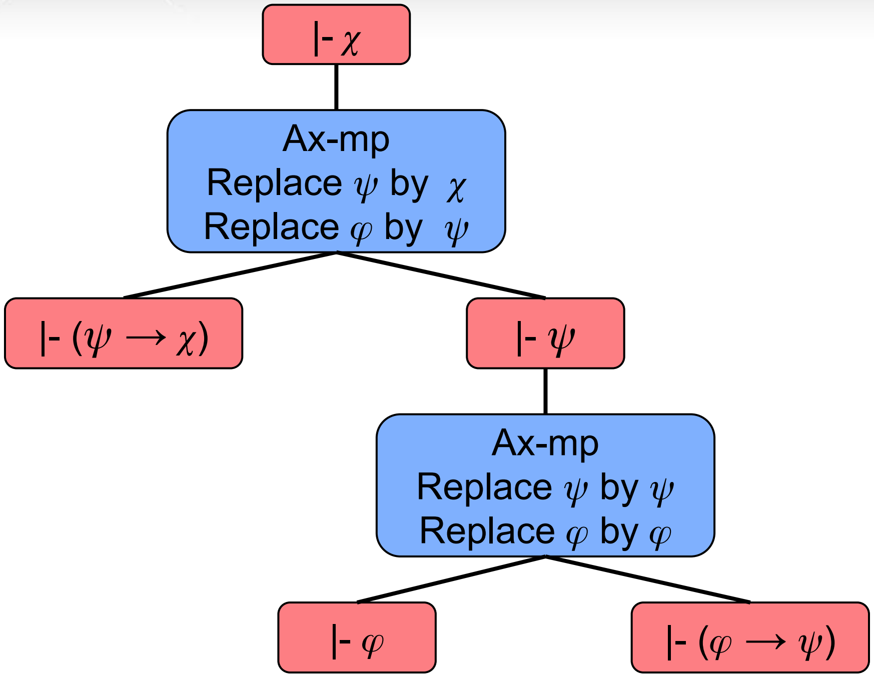
Ref	Expression
mp2b	$\vdash \chi$

Proof of Theorem mp2b

Step	Hyp	Ref	Expression
1		mp2b.1	..3 $\vdash \varphi$
2		mp2b.2	..3 $\vdash (\varphi \to \psi)$
3	1, 2	ax-mp 10	.2 $\vdash \psi$
4		mp2b.3	.2 $\vdash (\psi \to \chi)$
5	3, 4	ax-mp 10	1 $\vdash \chi$

fppt.com

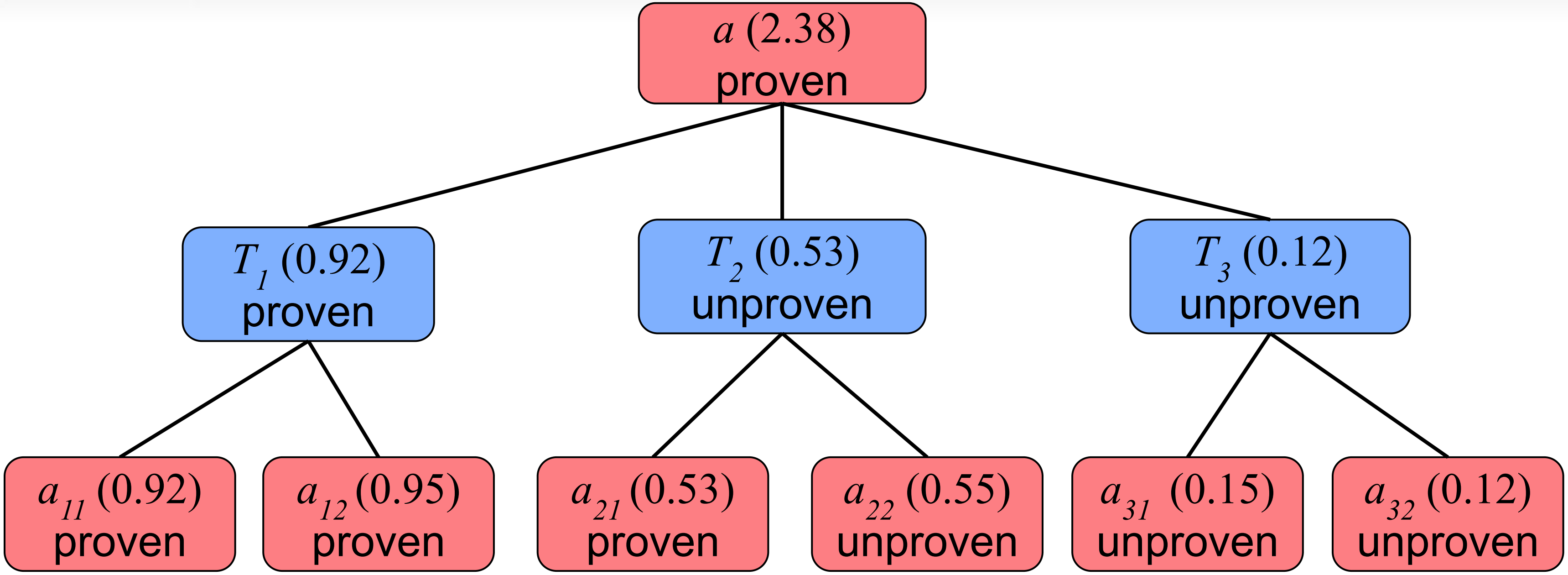
Constructing a Proof Tree



- A red node:
  - A statement
  - Has 0 or 1 child
- A blue node:
  - A theorem T
  - Parent: assertion $\phi(a_T)$
  - Children: hypotheses $\phi(e_T)$

fppt.com

Partial Proof Tree (PPT)



fppt.com

Partial Proof Tree (PPT)

- An expansion of a proof tree (if any)
- Thus, the PPT of a proven red node can be trimmed down to a standard proof tree.

fppt.com

Tree Search

- Suppose we want to prove a context C given:
  - its assertion a_C
  - A set of hypotheses

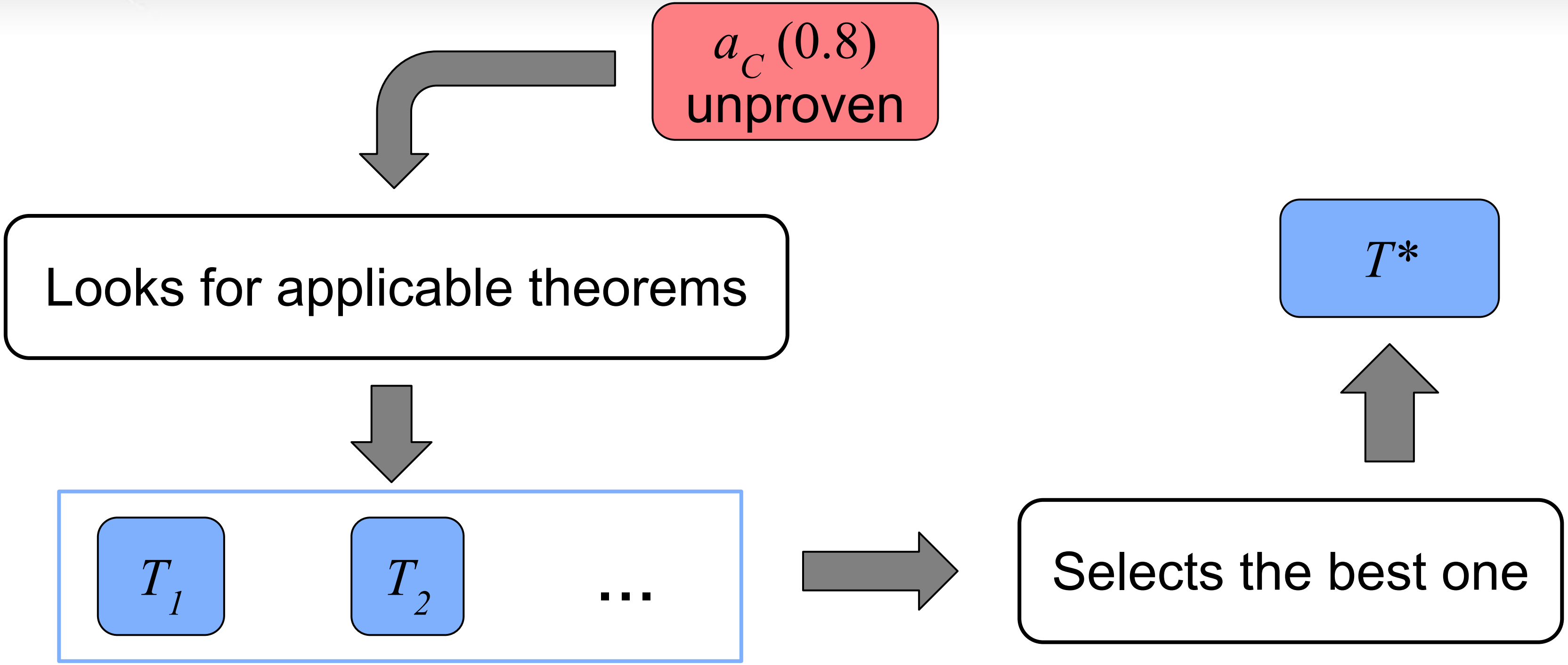
fpptt.com

Tree Search: First Pass

a_C (0.8)
unproven

fppt.com

Tree Search: First Pass



fppt.com

Tree Search: First Pass

a_C (0.8)
unproven

a_1 (1.0)

a_2 (0.8)

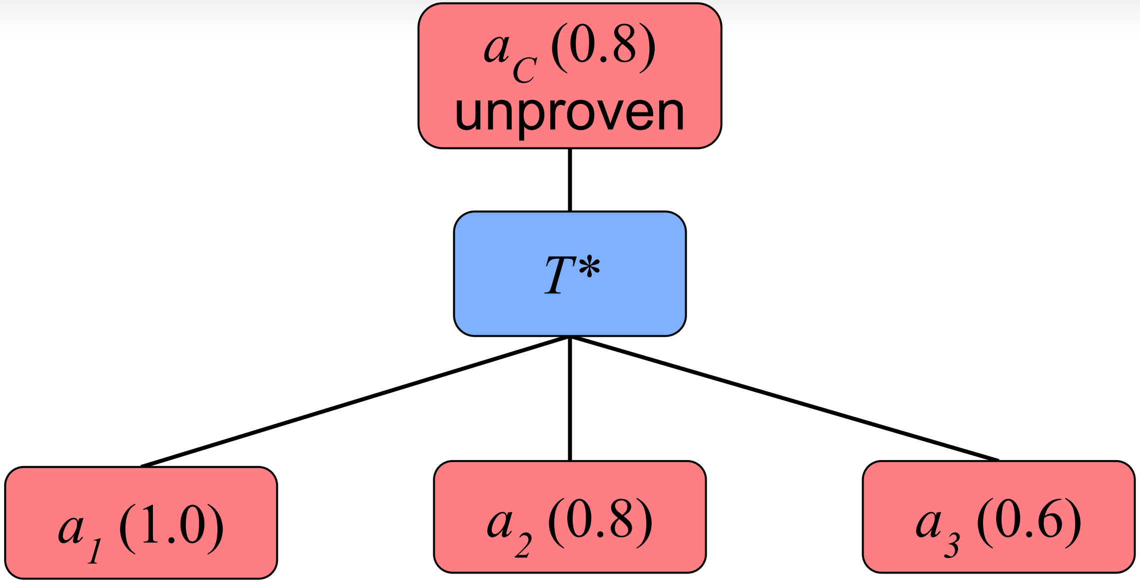
a_3 (0.6)

T^*

Determines suitable substitutions ϕ and returns $\phi(e_{T^*})$

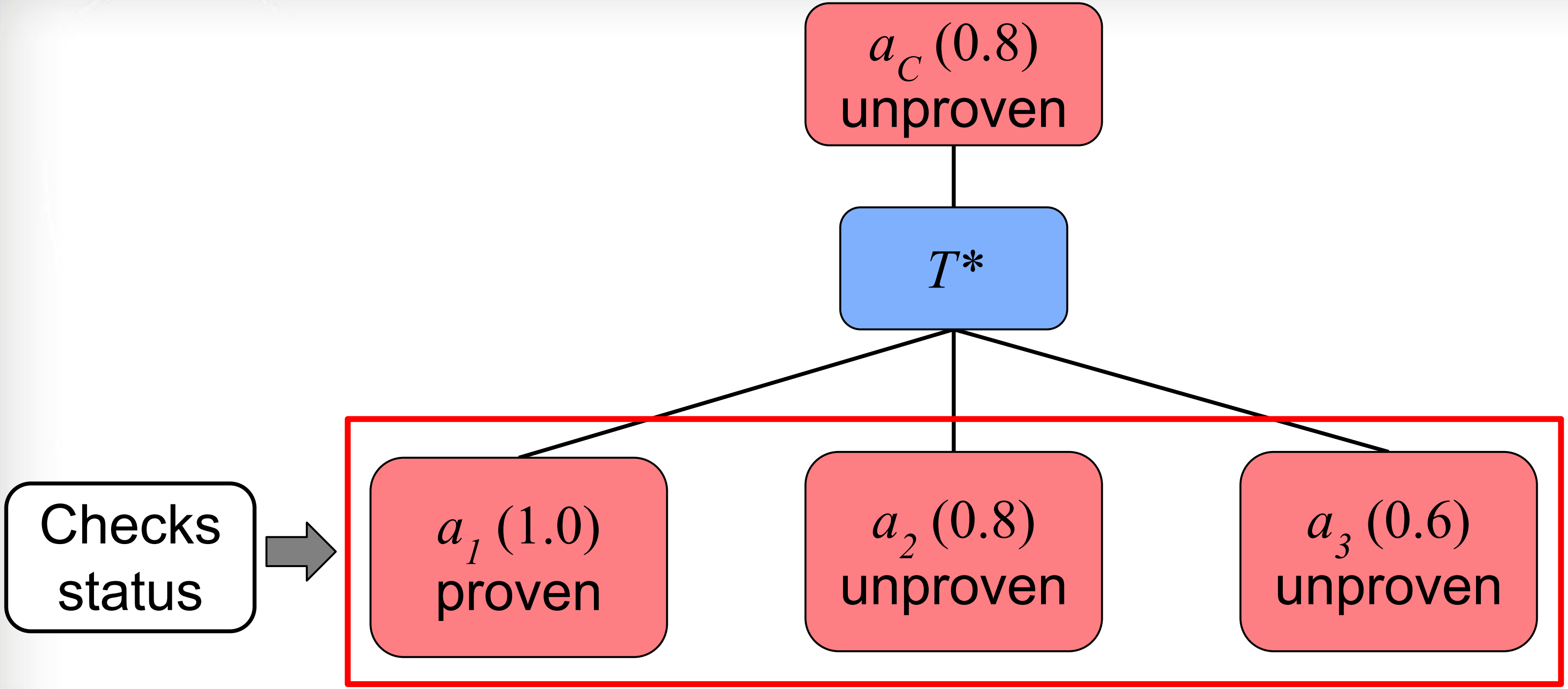
fppt.com

Tree Search: First Pass

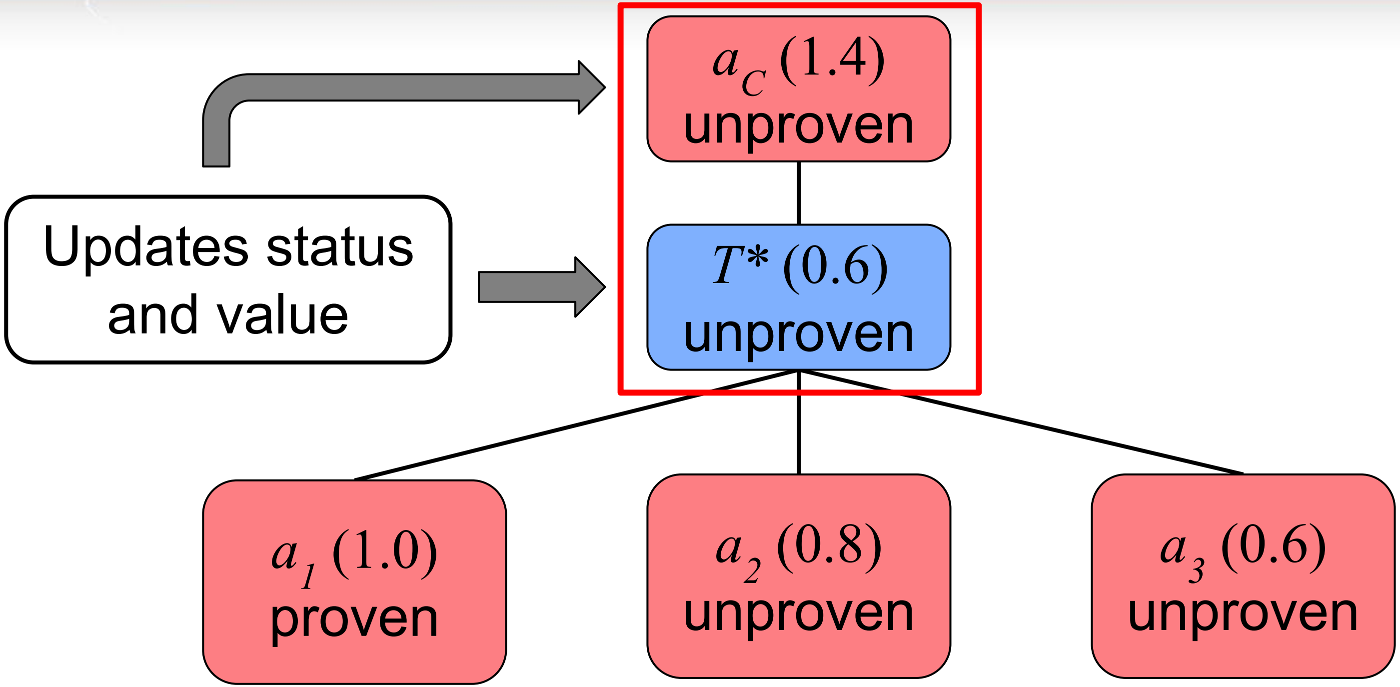


fppt.com

Tree Search: First Pass



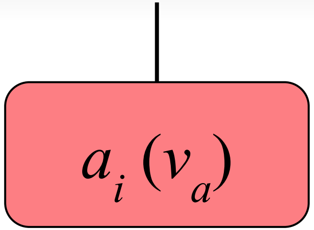
Tree Search: First Pass



Subsequent Passes: Red Nodes

No child blue node:

- Attempts to create one
- If cannot create one,
  - marks node as dead
  - ends current pass



fppt.com

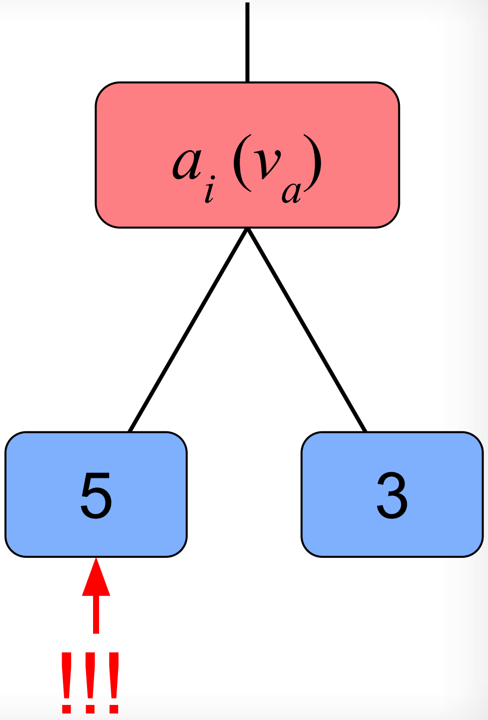
Subsequent Passes: Red Nodes

At least one child blue node:

- Creates a new child blue node;

or

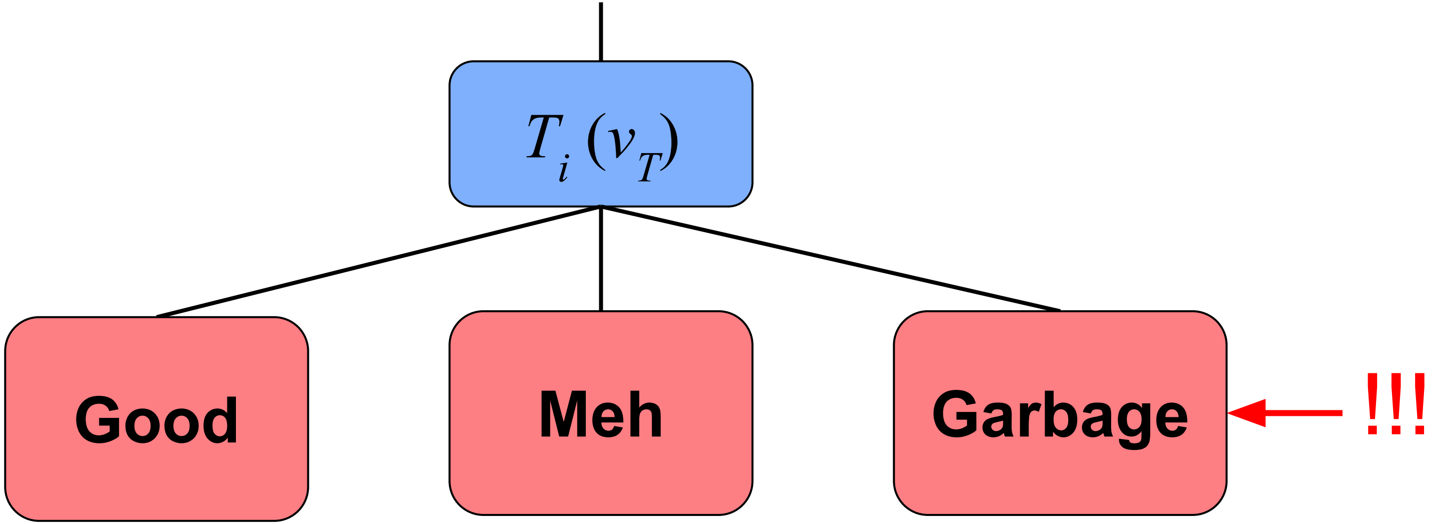
- Visits the child node with the <u>highest</u> score



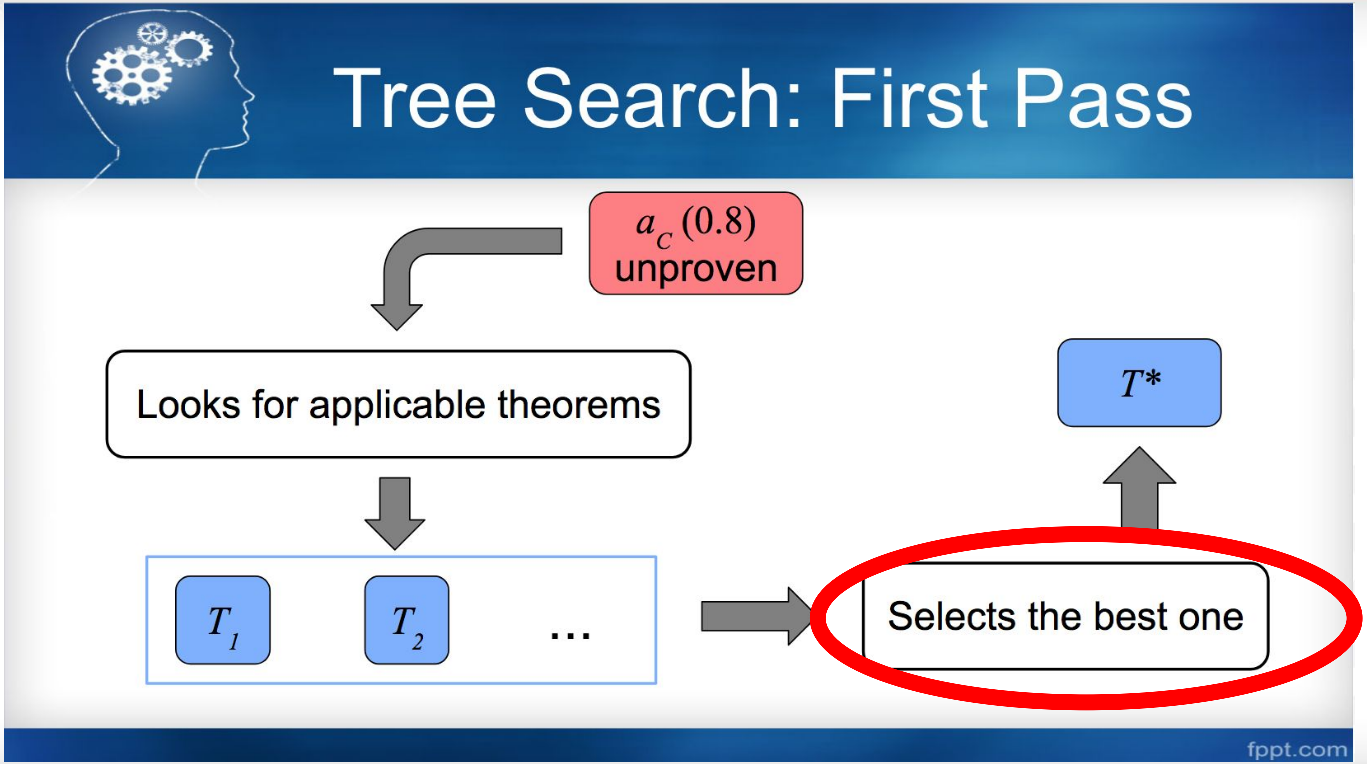
fppt.com

Subsequent Passes: Blue Nodes

Immediately visits its <u>worst</u> child red node



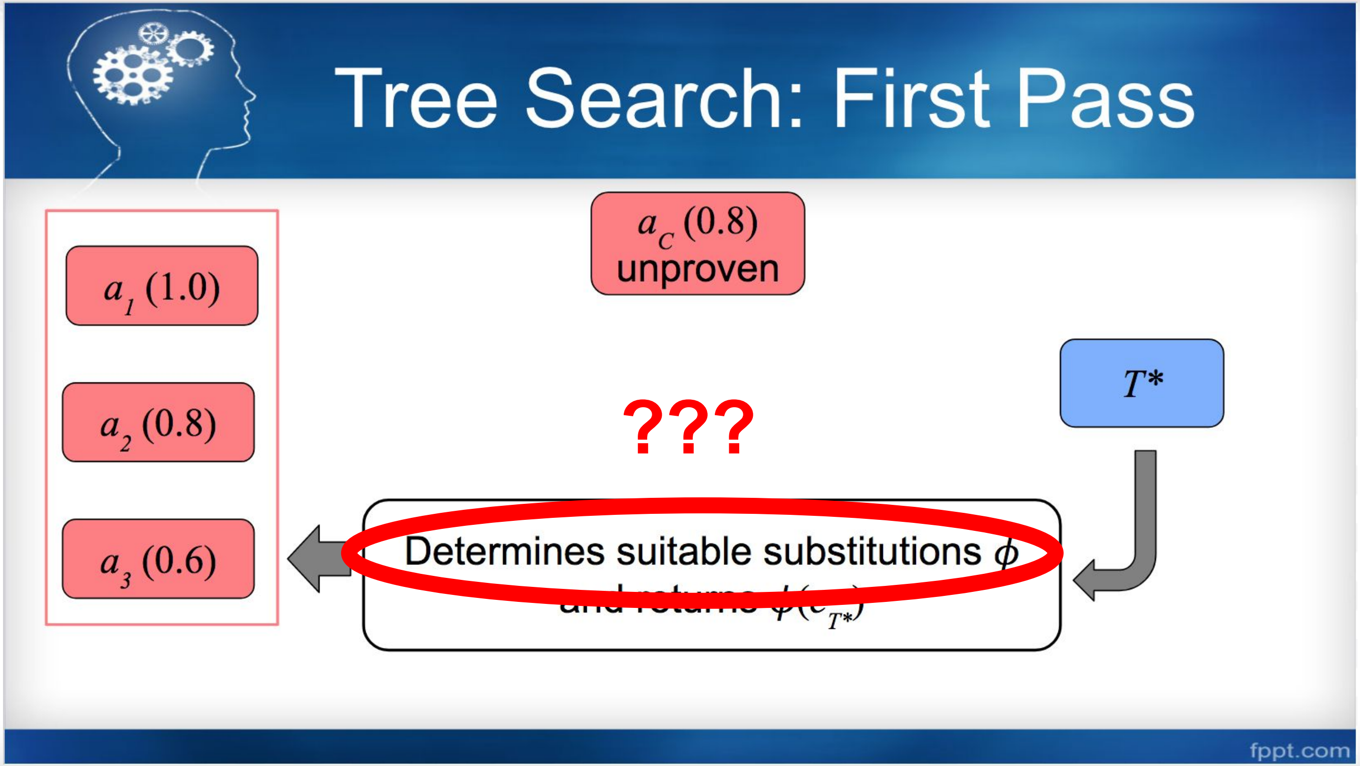
Challenges



Tree Search: First Pass
a_C (0.8)
unproven
Looks for applicable theorems
T_1
T_2
...
Selects the best one
T^*
fppt.com

???

Challenges



Tree Search: First Pass
a_1 (1.0)
a_2 (0.8)
a_3 (0.6)
a_C (0.8) unproven
???
T^*
Determines suitable substitutions ϕ
fppt.com

Challenges

Tree Search: First Pass

a_C (0.8)
unproven

???

fppt.com

Outline

- Introduction
- Holophrasm: The Program
  - Metamath Database
  - Proof Tree
  - The Neural Networks
- Improving Holophrasm: TFLearn
- Conclusion

fppt.com

Holophrasm

- Given a theorem, find a proof tree using deep learning.
- 3 neural networks
  - Relevance
  - Generative
  - Payoff
- All networks take assertions and hypotheses as inputs
  - Need to translate into vectors

fppt.com

Data preprocessing

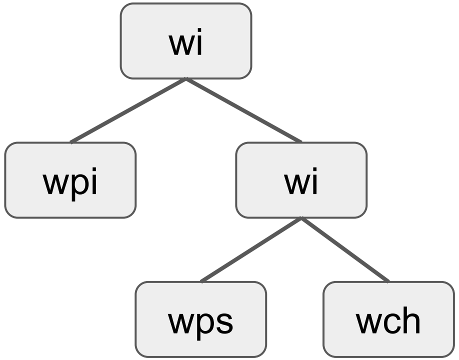
- All assertions and hypotheses have tree structure

Assertion

$$\varphi \rightarrow (\psi \rightarrow \chi)$$

Tree structure

wi(wpi(), wi(wps(), wch())

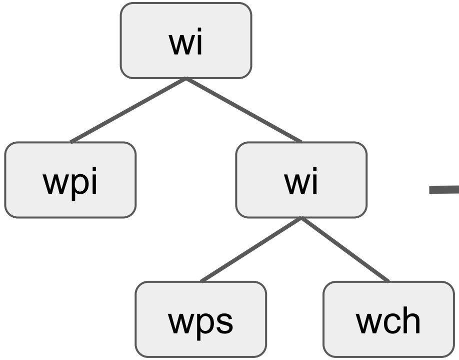


fppt.com

Data preprocessing

Tree structure

wi(wpi(), wi(wps(), wch()))



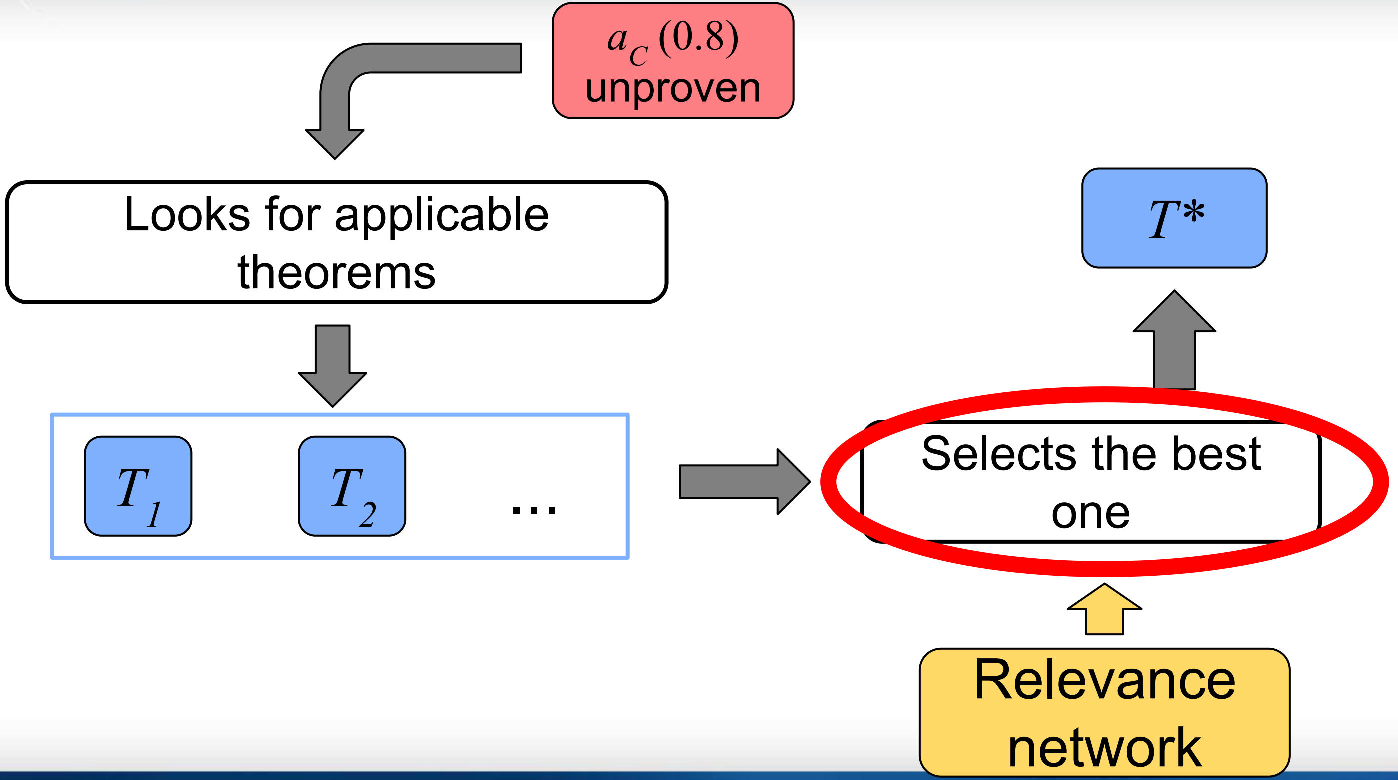
0	1	0	1	3
1	5	1	2	1
4	3	3	2	2
2	4	3	3	4
1	1	1	4	4
4	5	1	1	2
.	.	.	.	.
.	.	.	.	.
3	1	1	3	4
2	1	2	2	2
3	3	3	3	3
1	4	1	1	1
2	1	2	2	2

The content of a node

The structure data of a node

fppt.com

Relevance network

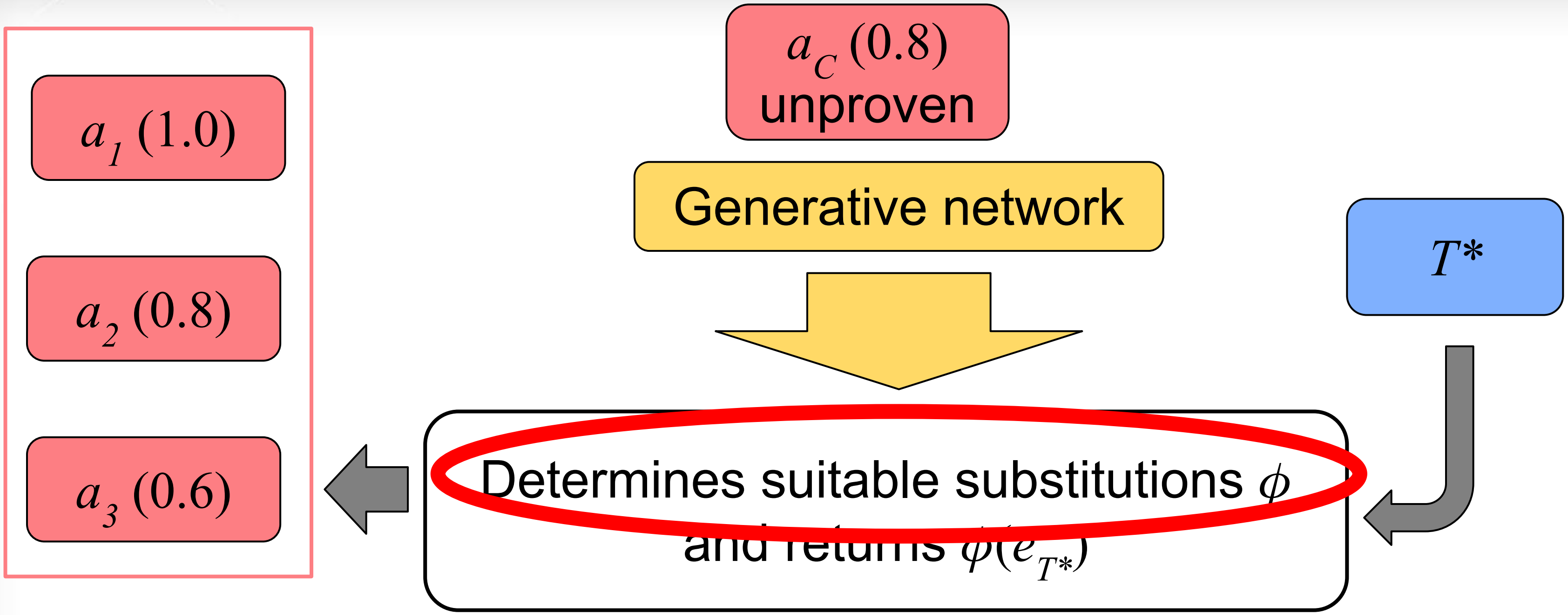


fppt.com

Relevance network

- Inputs
  - An assertion
  - A set of hypotheses of the context
- Output
  - Probabilities of theorems to be used in the node
- Uses RNN with GRU blocks

Generative network



Generative network

- Inputs
  - A set of hypotheses of a proposition
  - A set of hypotheses of the context
- Outputs
  - Set of substitutions
  - Probability of substitutions to be used
- Uses Sequence to Sequence Model with GRU blocks
  - Attention

fppt.com

Payoff network

$a_C(0.8)$
unproven

Payoff network

fppt.com

Payoff network

- Inputs
  - An assertion
  - A set of hypotheses of the context
- Outputs
  - Probability that assertion can be proved
- Uses RNN with GRU block
  - Bidirectional network

fppt.com

Outline

- Introduction
- Holophrasm: The Program
  - Metamath: A Formal Language
  - Proof Tree
  - The Neural Networks
- Improving Holophrasm: TFLearn
- Conclusion

fppt.com

TensorFlow

- He used his own library
  - Difficult to understand
- Need to replace the code with well-known library!

```
def GRUbCell(hin, x, GRUparams, graph, dropout=None):
    hx = ConcatNode([hin, x], graph)
    # print hin.value.shape, x.value.shape
    # print hx.value.shape, GRUparams.Wz.value.shape
    z = DotNode(hx, GRUparams.Wz, graph)
    z = AddNode([z, GRUparams.bz], graph)
    z = SigmoidNode(z, graph)
    r = DotNode(hx, GRUparams.Wr, graph)
    r = AddNode([r, GRUparams.br], graph)
    r = SigmoidNode(r, graph)
    rh = TensorMultiplyNode(r, hin, graph)
    rhx = ConcatNode([rh, x], graph)
    htilde = DotNode(rhx, GRUparams.W, graph)
    htilde = TanhNode(htilde, graph)
    htilde = DropoutNode(htilde, dropout, graph)

    negz = MultiplyNode(-1.0, z, graph)
    negz = ScalarAddNode(1, negz, graph)

    negzh = TensorMultiplyNode(negz, hin, graph)
    zhtilde = TensorMultiplyNode(z, htilde, graph)

    hout = AddNode([negzh, zhtilde], graph)

    return hout
```

fppt.com

Companies using TensorFlow

airbnb
AIRBUS DEFENCE & SPACE
ARM
CEVA
CI&T
DeepMind
Dropbox
ebay
Google
IBM Power Systems
intel
JD.COM 京东
mi
Movidius
QUALCOMM
quantiphi
SAP
UBER

fppt.com

Build History

trend

find

#259	Aug 2, 2017 2:25 AM
#258	Aug 1, 2017 2:25 AM
#257	Jul 31, 2017 12:38 PM
#256	Jul 31, 2017 2:25 AM
#255	Jul 30, 2017 2:25 AM
#254	Jul 29, 2017 2:25 AM
#253	Jul 28, 2017 2:25 AM
#252	Jul 27, 2017 2:25 AM
#251	Jul 26, 2017 2:25 AM
#250	Jul 25, 2017 2:25 AM
#249	Jul 24, 2017 2:25 AM
#248	Jul 23, 2017 2:25 AM
#247	Jul 22, 2017 2:25 AM
#246	Jul 21, 2017 2:25 AM
#245	Jul 20, 2017 2:25 AM
#244	Jul 19, 2017 2:25 AM
#243	Jul 18, 2017 2:25 AM
#242	Jul 17, 2017 2:25 AM
#241	Jul 16, 2017 4:35 AM
#240	Jul 16, 2017 12:20 AM
#239	Jul 15, 2017 2:25 AM
#238	Jul 14, 2017 2:25 AM
#237	Jul 13, 2017 2:25 AM
#236	Jul 12, 2017 2:25 AM
#235	Jul 11, 2017 2:25 AM

stent

ble m

adversarial_crypto	Comment wrong in adversarial_crypto model	a month ago
adversarial_text	Fix KL when num_classes != 2 (#1820)	a month ago
attention_ocr	Add ./ to --checkpoint_inception (#1951)	16 days ago
autoencoder	Variational Autoencoder generate() function fixed (z fed in rather th...	4 months ago
cognitive_mapping_and_planning	Update README.md	18 days ago
compression	Update username and add new line in READMEs.	2 months ago
differential_privacy	typo fix	2 months ago
domain_adaptation	Open sourcing PixelDA code	13 days ago
im2txt	Fix formatting	15 days ago
inception	Fixed error message for inception/imagenet.	2 months ago
learning_to_remember_rare_events	Convert tf.op_scope to tf.name_scope, plus a few other 1.0 upgrade ch...	5 months ago
lfads	Removing redundant print statement	6 days ago
lm_1b	Import xrange directly from six.moves for lm_1b	3 months ago
namignizer	Merge pull request #924 from h4ck3rm1k3/master	5 months ago
neural_gpu	Fix arxiv links in README of neural_gpu model	2 months ago
neural_programmer	Merge pull request #924 from h4ck3rm1k3/master	5 months ago
next_frame_prediction	Improvements to several READMEs	3 months ago
object_detection	Set training step limits on pets configs.	5 days ago
pcl_rl	code for running RL agents with various algs	28 days ago
ptn	Added training details comment.	14 days ago
real_nvp	change no-longer-existing concat_v2 to concat (#1701)	a month ago
rebar	Update README.md	9 days ago
resnet	Offset of image bytes for cifar100 should be 2	9 days ago
skip_thoughts	Fix Bazel incantations in docs	2 months ago
slim	Merge pull request #1800 from txizzle/master	6 days ago

TensorFlow

- GPU, Multithreading, Distributed training

```
train_object.run_many_epochs(language_model, plot_every=10000,
# Creates a session with log_device_placement set to True.
sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
                        batch_size=100, multiprocessing=8,
                        reorder=False)
```

```
sess = tf.Session(config=tf.ConfigProto(
  intra_op_parallelism_threads=NUM_THREADS))
```

```
                                                  process before the
        current process has finished its bootstrapping phase.

        This probably means that you are not using fork to start your
        child processes and you have forgotten to use the proper idiom
```

```
>>> server = tf.train.Server.create_local_server()
>>> sess = tf.Session(server.target)  # Create a session on the server.
```

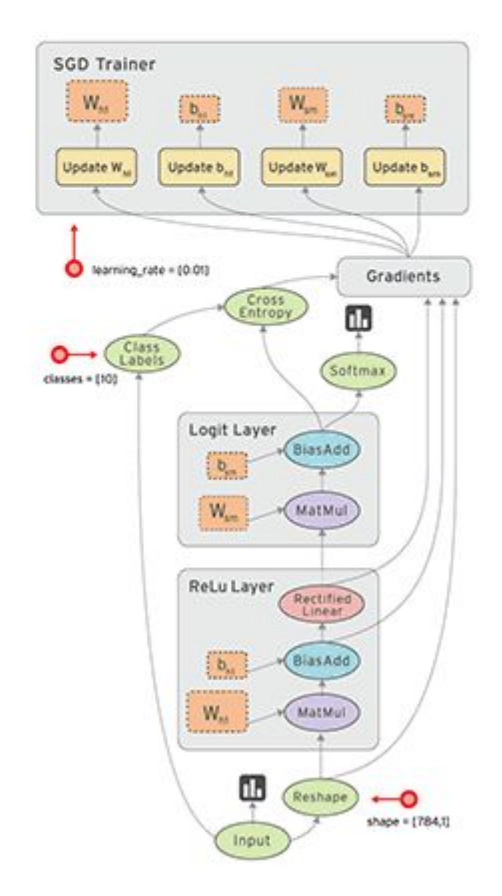
```
            if __name__ == '__main__':
                freeze_support()
                ...

        The "freeze_support()" line can be omitted if the program
        is not going to be frozen to produce an executable.
```

fppt.com

TensorFlow

- TensorBoard
  - Visualization of Deep Learning
  - Graph, Loss decrease, etc.



fppt.com

TensorFlow

- Unfortunately, we are all beginners in developing deep learning.
- Even TensorFlow is too difficult to use!
- We don’t have time!
- We need easier one.

fppt.com

TFLearn

- Developed by Aymeric Damien, available at tflearn.org
- Source code : Github.com/tflearn/tflearn
- Higher-level API of TensorFlow

fppt.com

TFLearn

- Very easy, short, and intuitive

```
main_net = tflearn.fully_connected(main_net, output_dim, weights_init='xavier')
main_net = tflearn.relu(main_net)
main_net = tflearn.fully_connected(main_net, output_dim, weights_init='xavier')
main_net = tflearn.relu(main_net)
main_net = tflearn.dropout(main_net, p)
```

Layers

- Core Layers
- Convolutional Layers
- Recurrent Layers
- Normalization Layers
- Embedding Layers
- Merge Layers
- Estimator Layers

fppt.com

TFLearn

- Automatically detects GPU

```
# Creates a session wi[illegible]to True.
sess = tf.Session(confi[illegible]placement=True))
```

No extra code for using GPU!

TFLearn

- Easy construction of TensorBoard

```
# Create other useful summary (weights, grads, activations...)
# according to 'tensorboard_verbose' level.
self.create_summaries(tensorboard_verbose)
```

TFLearn library built-in code

```
model = tflearn.DNN(net, tensorboard_verbose=3)
```

What we need for constructing Tensorboard

fppt.com

TFLearn

- Full transparency over TensorFlow
  - If you want models not in TFLearn, you can build it with TensorFlow and TFLearn together!
  - Example : https://github.com/tflearn/tflearn/blob/master/examples/nlp/seq2seq_example.py

fppt.com

TFLearn

- We implemented TFLearn into Relevance network! (but incomplete)
- 1302 lines of 4 codes -> 452 lines in 2 codes

406
407
408

54
55
56

613
614
615

223
224
225

252
253
254

196
197
198

fppt.com

Relevance network

- It is basically a classification problem
- But too many propositions(classes)
  - Huge computational time

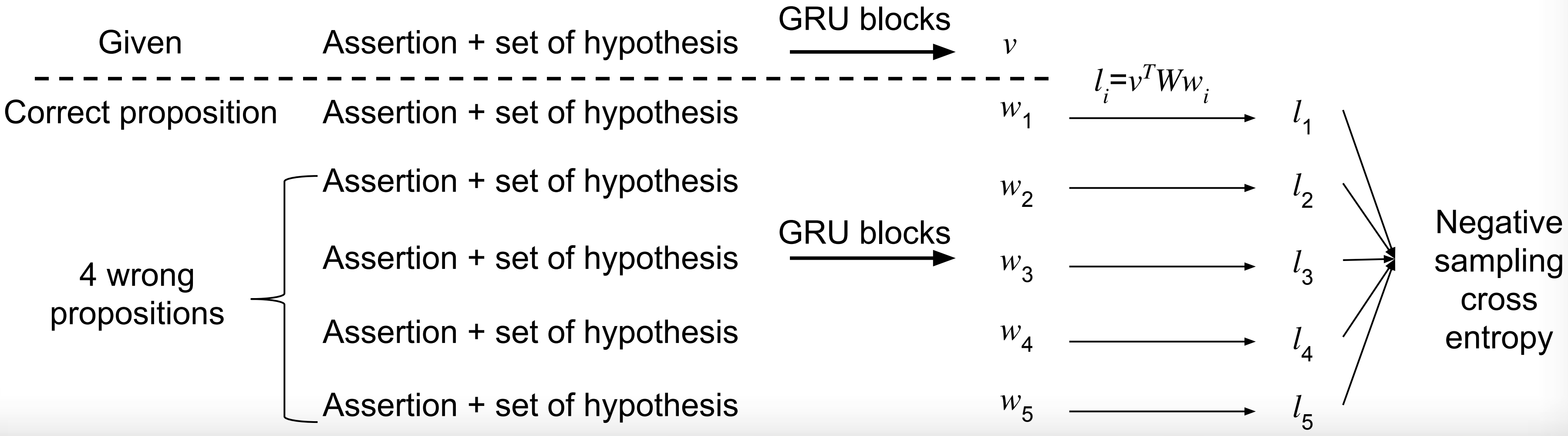
→ Cat

Hypotheses	Assertion
$\vdash \varphi$	$\vdash \chi$
$\vdash \varphi \to \psi$	
$\vdash \psi \to \chi$	

→ Modus ponens

Relevance network

- Negative sampling technique was used to minimize the computational cost
  - Pick 1 correct proposition and 4 wrong viable propositions



fppt.com

Data preprocessing

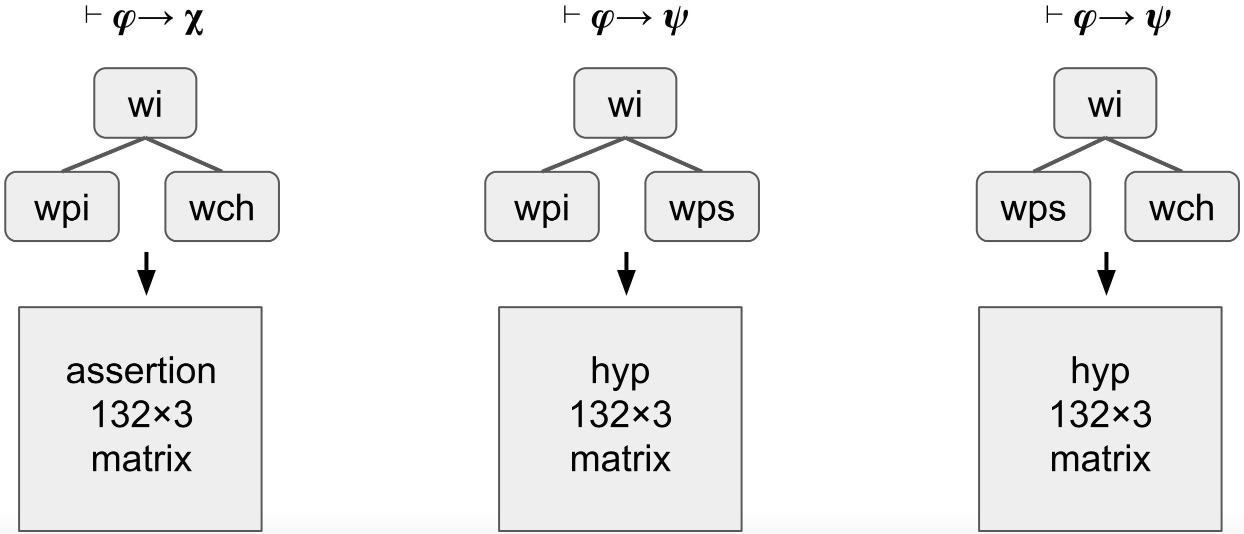
Assertion

$\vdash \varphi \rightarrow \chi$

Hypotheses

$\vdash \varphi \rightarrow \psi$

$\vdash \psi \rightarrow \chi$



fppt.com

Data preprocessing

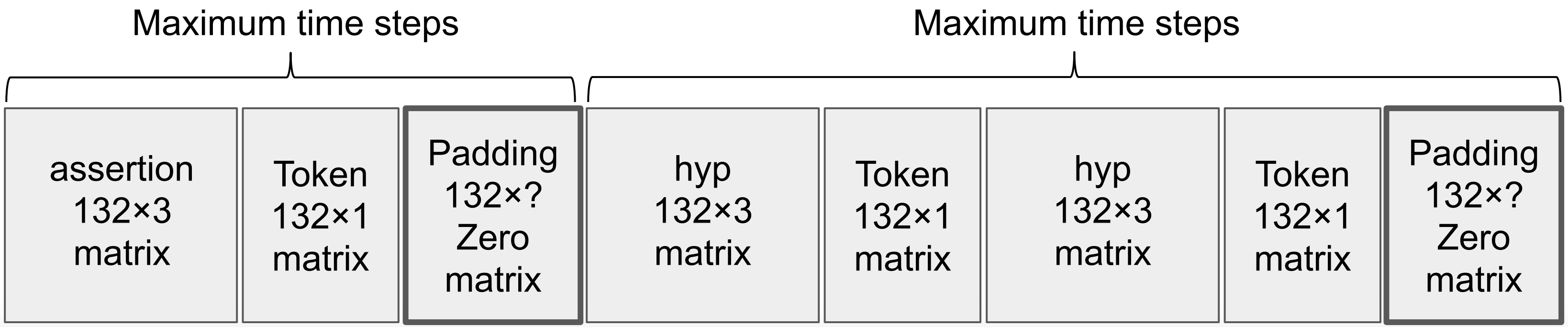
- Tokens are added between assertion and hypotheses
  - All entries are -1
- All matrices are concatenated
  - Row size is flexible

assertion 132×3 matrix	token 132×1 matrix	hyp 132×3 matrix	token 132×1 matrix	hyp 132×3 matrix	token 132×1 matrix

fppt.com

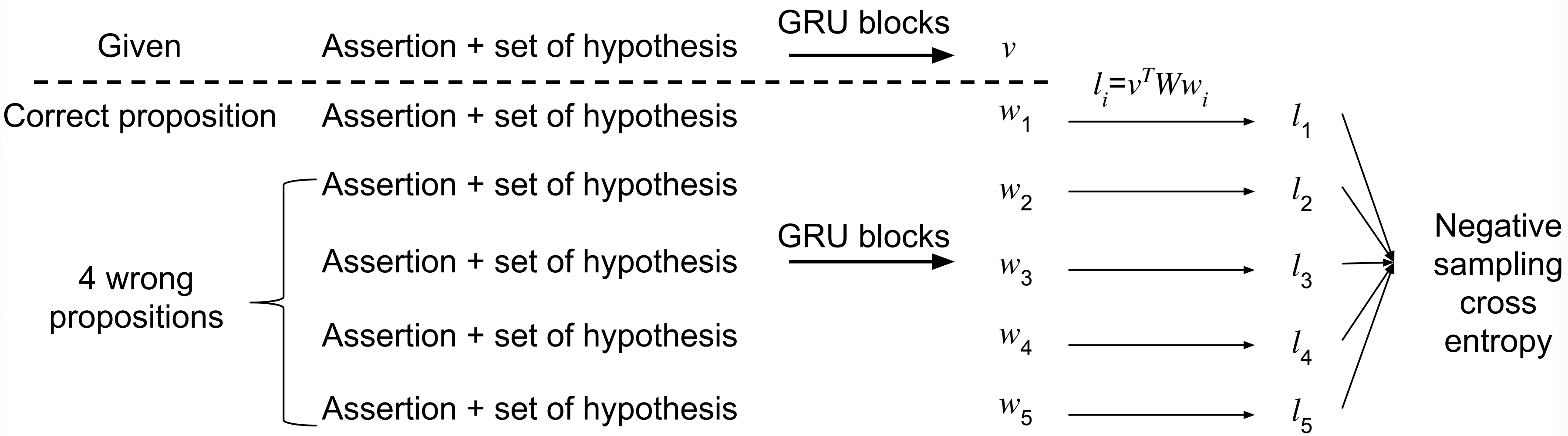
Data preprocessing

- Each column vector will be fed into one GRU Block
  - The number of GRU blocks we need varies
- To use **dynamic** RNN in TFLearn, we have to make all inputs have the same row length.
- Two zero matrices are padded



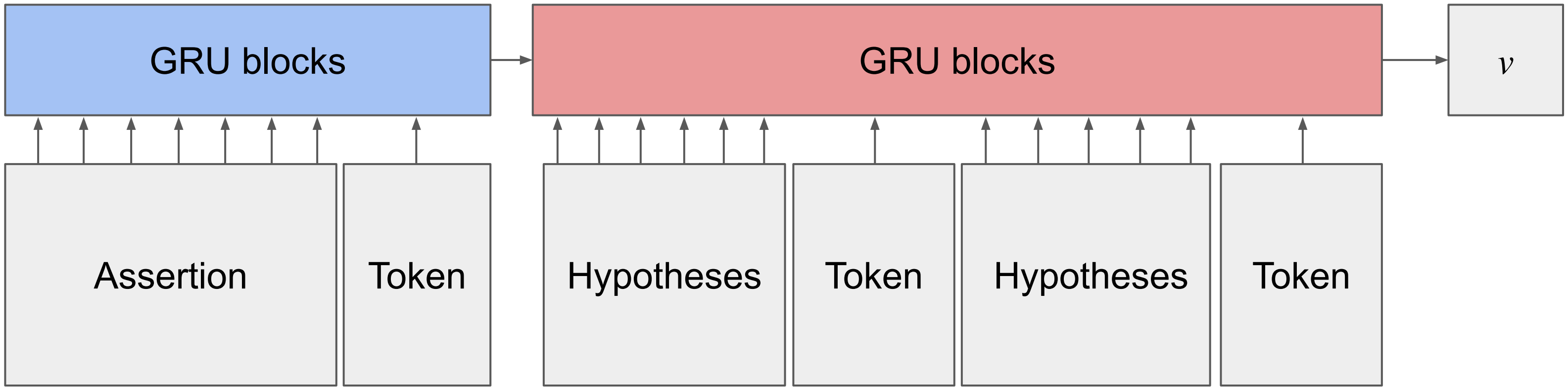
fppt.com

Creating the network



The first network

Given Assertion + set of hypothesis $\xrightarrow{\text{GRU blocks}} v$



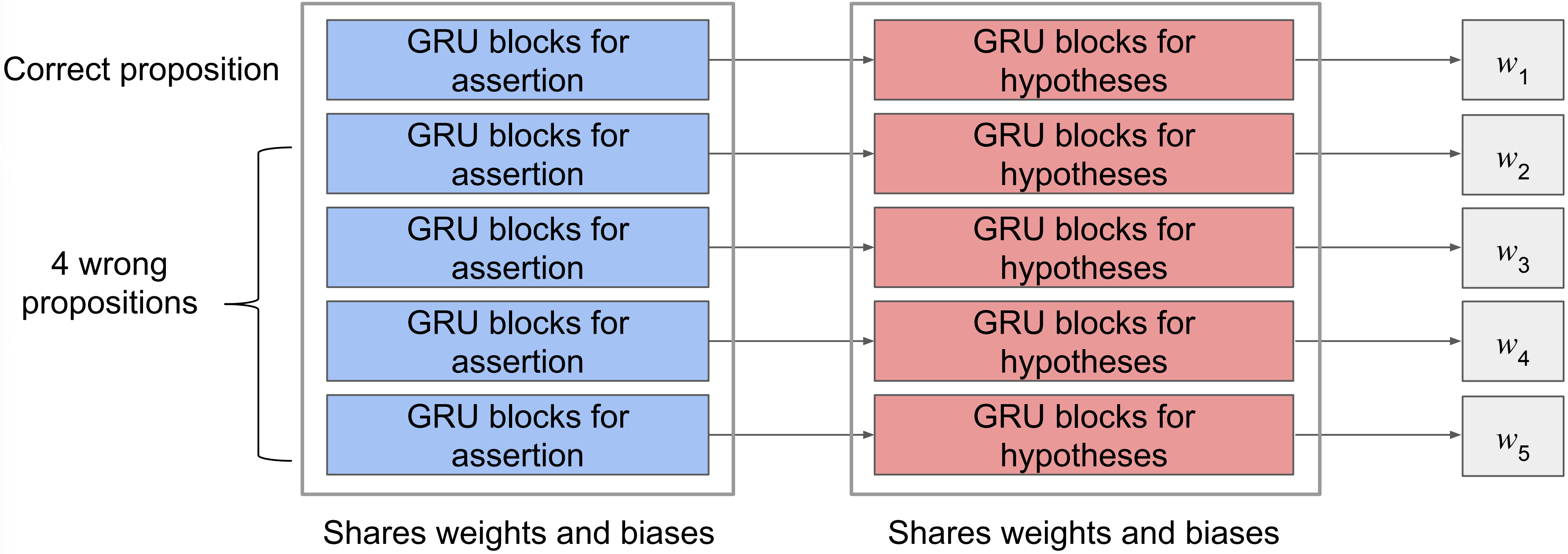
fppt.com

The second network

Correct proposition	Assertion + set of hypothesis		w_1
4 wrong propositions	Assertion + set of hypothesis		w_2
	Assertion + set of hypothesis	GRU blocks $\longrightarrow$	w_3
	Assertion + set of hypothesis		w_4
	Assertion + set of hypothesis		w_5

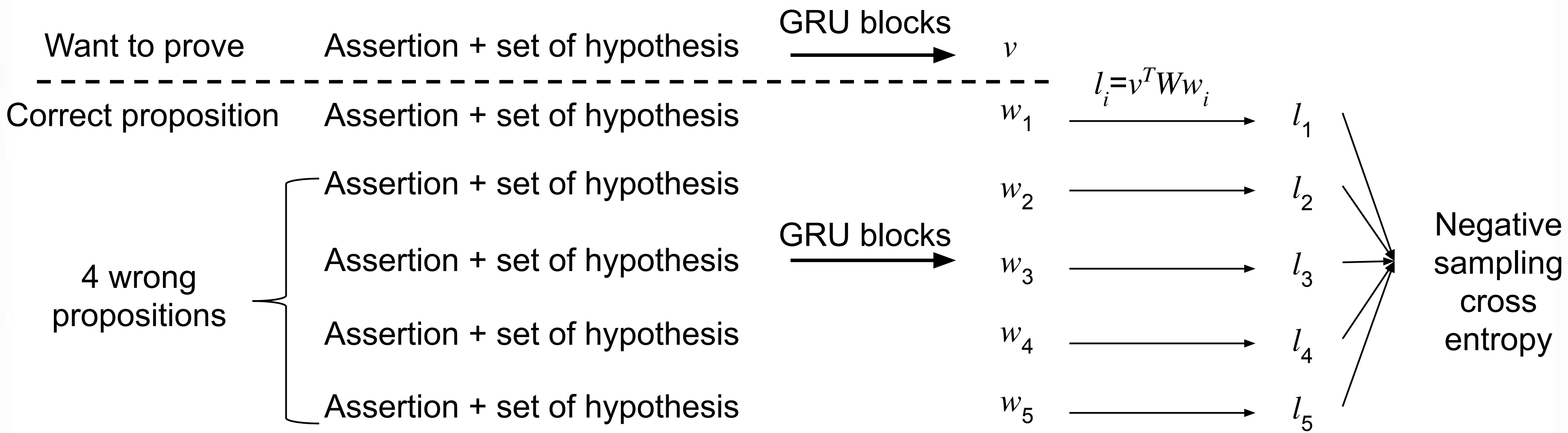
fppt.com

The second network



fppt.com

Creating the network



fppt.com

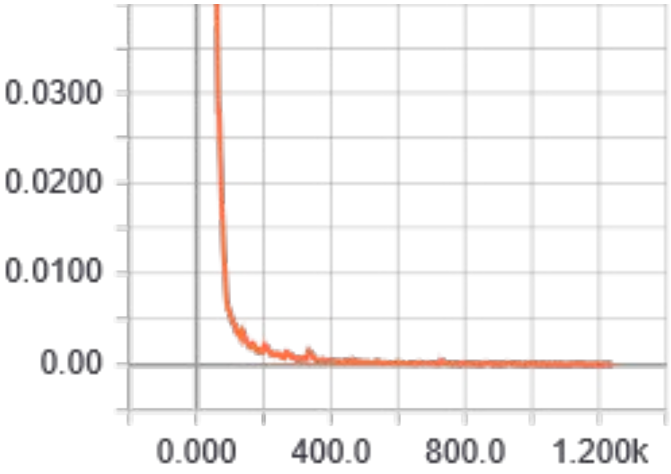
Training the network

- Uses the first 2000 propositions in Metamath database.
- 6480 training data are extracted from the propositions.
- Batch size = 100
- 20 epochs

fppt.com

Results

- The loss function decreases well
- We don't have enough time to test the network.
  - Need to construct a new network to test (negative sampling)



Outline

- Introduction
- Holophrasm: The Program
  - Metamath Database
  - Proof Tree
  - The Neural Networks
- Improving Holophrasm: TFLearn
- Conclusion

fppt.com

Conclusion

- Holophrasm: an automated theorem prover
  - Searches for a proof tree within a PPT
  - Guided by three neural networks
- Improved Holophrasm with TFLearn

fppt.com

Future Work

- Replace generative and payoff networks

- Test and improve

fppt.com

References

Norman Megill. "Metamath: A Computer Language for Pure Mathematics" (PDF). Lulu Press, Morrisville, North Carolina. ISBN 978-1-4116-3724-5. p. xi

Daniel P.Z. Whalen(2016). “Holophrasm: a neural Automated Theorem Prover for higher-order logic”. : arxiv:1608.02644

Norman Megill. “Demo0.mm” Jan 1, 2004, Retrieved July 13, 2017

"Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU/LSTM RNN with Python and Theano – WildML". *Wildml.com*. Retrieved July 13, 2017.
“Accelerating Deep Learning Research with the Tensor2Tensor Library”. *Research.googleblog.com*. Retrieved July 14, 2017.

“TensorFlow”, *TensorFlow.org.* Retrieved July 14, 2017

“TFLearn”, *TFlearn.org*. Retireved August 3, 2017

fppt.com

Thank you for listening!

fppt.com

Thank you for listening!

Any questions?

fppt.com