LTN Backgroun

LTN Within NSI

LIN Evolution

LTN Component

LTN Real Logic Axid

LTN Knowledge Base

LTN Loss Function

LTN Training

LTN Toy Example

Github Repo Examples
Binary Classification

LTN Academic Us

Summar

# Introduction to the Logic Tensor Networks (LTN) Framework

Assembled for PhD Students of Ernesto & Tillman Interested in Neural-Symbolic Integration (NSI)

Dave Herron

City, University of London

December 2021



#### Introduction

LTN Background

LTN VICINIII

Logic Revi

LTN Component

LTN D. II.

LTN Real Logic Axid

LTN Knowledge Bas

LTN Loss Fun

LTN Training

Cithuh Pone Example

Binary Classification

LTN Academic Use

Summar

## What we're going to talk about

see Navigation bar

#### Introduction

LTN Background

LTN VVILINI IV

LTN.C

LTN Componer

LTN Real Lo

LTN Real Logic Axio

LTN Knowledge Base

LIN LOSS FUI

LTN Trainin

City Bara Everal

Binary Classification

LTN Academic Us

Summar

## What we're going to talk about

see Navigation bar

#### Introduction

LTN Backgroun

LTN Within NSI

Logic Revi

LTN Component

LTTV Componer

LIN Real Log

LTN Kear Logic And

LTN Loss Function

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Us

Summar

### What we're going to talk about

see Navigation bar

### Key messages:

► LTN is a **framework** for combining **prior knowledge** with **data** in the training of **conventional** neural networks

#### Introduction

- LTN Backgroun
- LTN Within NSI
- Logic Revie
- LTN Componer
- \_\_\_\_\_
- ETTA IVEST LOGI
- LTN K L L D
- LTN Loss Funct
- LTN Toy Eva
- Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Us

Summar

### What we're going to talk about

▶ see Navigation bar

- ► LTN is a **framework** for combining **prior knowledge** with **data** in the training of **conventional** neural networks
- ► LTN is **not** a special **type** of neural network (NN)

#### Introduction

LTN Backgroun

LTN Within NSI

Logic Revie

LTN Componen

LTN Real Logic Axio

LTN Loss Eugstion

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

C...mm.nv

### What we're going to talk about

see Navigation bar

- ► LTN is a **framework** for combining **prior knowledge** with **data** in the training of **conventional** neural networks
- ► LTN is **not** a special **type** of neural network (NN)
- ► knowledge is expressed as logical axioms over data using a fuzzy, first-order logic language

#### Introduction

- LTN Backgroun
- LTN Within NSI
- Logic Revie
- LTN Component
- LTN Real Logic
- LTN Real Logic
- LTN Knowledge Base
- LTN Loss Fund
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification

LTN Academic Use

Summar

### What we're going to talk about

see Navigation bar

- ► LTN is a **framework** for combining **prior knowledge** with **data** in the training of **conventional** neural networks
- ► LTN is **not** a special **type** of neural network (NN)
- knowledge is expressed as logical axioms over data using a fuzzy, first-order logic language
- neural networks are predicates in the knowledge axioms

#### Introduction

- LTN Backgroun
- LTN Within NSI
- Logic Revie
- LTN Componer
- LTN Real Lori
- LTN Real Logic
- LTN Knowledge Base
- LTN Loss Funct
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Use Summary

### What we're going to talk about

see Navigation bar

- ► LTN is a **framework** for combining **prior knowledge** with **data** in the training of **conventional** neural networks
- ► LTN is **not** a special **type** of neural network (NN)
- knowledge is expressed as logical axioms over data using a fuzzy, first-order logic language
- neural networks are predicates in the knowledge axioms
- conventional loss minimisation drives the satisfiability of the knowledge axioms which, in turn, drives the training of the neural network

#### Introduction

- LTN Backgroun
- LTN Within NSI
- Logic Revie
- LTN Componer
- LTN Real Log
- LTN Post Logi
- LTN Knowledge Rass
- LTN Loss Funct
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Use Summary

### What we're going to talk about

► see Navigation bar

- ► LTN is a **framework** for combining **prior knowledge** with **data** in the training of **conventional** neural networks
- ► LTN is **not** a special **type** of neural network (NN)
- knowledge is expressed as logical axioms over data using a fuzzy, first-order logic language
- neural networks are predicates in the knowledge axioms
- conventional loss minimisation drives the satisfiability of the knowledge axioms which, in turn, drives the training of the neural network
- ▶ the optimal function the NN learns is the one that best satisfies the knowledge axioms (given the data)

## Context of LTN re Neural-Symbolic Integration

Introduction

LTM Rackgroup

LTN Within NSI

. . . . .

LTNG

LTN Componer

LTN Real Log

LTN Real Logic Axio

LTN Loss Eunstion

LTN Training

LTN Toy Example

Binary Classification
Clustering

LTN Academic Us

Summar

## Neural-symbolic integration (NSI)

- combining connectionist AI with symbolist AI
- several classes of approach have emerged

## Context of LTN re Neural-Symbolic Integration

Introduction

TN Backgroun

LTN Within NSI

Lauta Basil

LTN Componen

LIN Componer

LTN Real Logic

LTN Knowledge Base

LTN Training

LIN Iraining

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Us

Summar

## Neural-symbolic integration (NSI)

- combining connectionist AI with symbolist AI
- several classes of approach have emerged

## One class integrates learning and reasoning

blending neural network deep learning with logical reasoning in a fully-differentiable, end-to-end architecture

## Context of LTN re Neural-Symbolic Integration

Introduction

TN Backgrour

LTN Within NSI

ogic Rev

LTN Componer

LTN Real Logic

LTN Knowledge Base

LTN Loss Functi

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Us Summary

## Neural-symbolic integration (NSI)

- combining connectionist AI with symbolist AI
- several classes of approach have emerged

### One class integrates learning and reasoning

blending neural network deep learning with logical reasoning in a fully-differentiable, end-to-end architecture

### One **subclass** of this class

- measures the satisfiability of logical knowledge and weaves this into the loss function
- ► LTN is one member of this subclass

#### Introduct

LTN Backgroun

#### LTN Evolution

ogic Review

LTN Component

LTN D. II.

LTN Real Logic Axio

LTN Knowledge Base

LIN LOSS FUI

LTN Training

LTN Toy Example

Binary Classification
Clustering

LTN Academic Us

Summary

## Versions of LTN

## Inspiration

 "Reasoning with Neural Tensor Networks for Knowledge Base Completion"; Socher, Chen, Manning & Ng, 2013.

## (LTN) Framework

## LTN Evolution

## Versions of LTN

## Inspiration

"Reasoning with Neural Tensor Networks for Knowledge Base Completion": Socher, Chen, Manning & Ng. 2013.

## Original version (2016)

- "Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge"; Serafini & Garcez, 2016
- TensorFlow 1 (explicit, static computational graphs)

## Versions of LTN

#### Introduc

LTN Backgroun
LTN Within NSI
LTN Evolution

#### Logic Rev

LTN Componen

LTN Real Logic

LTN Real Logic

LTN Loss Functi

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

Summary

## Inspiration

 "Reasoning with Neural Tensor Networks for Knowledge Base Completion"; Socher, Chen, Manning & Ng, 2013.

## Original version (2016)

- "Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge"; Serafini & Garcez, 2016
- ► TensorFlow 1 (explicit, static computational graphs)

## New version (early 2021)

- ► "Logic Tensor Networks"; Badreddine et al., 2021
- ► TensorFlow 2 (implicit, dynamic computational graphs)
- extended features; gradient stability enhancements
- https://github.com/logictensornetworks/logictensornetworks
  - LTN API code, tutorials, examples, link to 2021 paper
  - this presentation borrows from these assets!



## Classical vs Fuzzy Logics

#### Introduction

.TN Backgroun

#### Logic Review

LTN Componen

LTN Real Lo

LTN Real Logic Axio

LTN Knowledge Base

LIN LOSS Fun

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification

LTN Academic Us

Summan

### Classical logics

- assume **precision** (e.g. John is a person)
- **binary** truth values:  $\{0,1\}$  (False or True)
- main types: propositional logic; first-order logic

## Classical vs Fuzzy Logics

#### Introduction

- LTN Backgrour
- LTN Within NSI

#### Logic Review

- LTN C----
- LTN Post Log
- LTN Real Logic Axion
- LTN Knowledge Base
- LTN Loss Fun
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Us

Summar

### Classical logics

- assume **precision** (e.g. John is a person)
- **binary** truth values:  $\{0,1\}$  (False or True)
- main types: propositional logic; first-order logic

### **Fuzzy logics**

- arose to handle vagueness (e.g. John is tall)
- ▶ infinite degrees of truth:  $\in [0,1]$
- main types: fuzzy propositional logic; fuzzy first-order logic
- major fuzzy logic systems: Łukasiewicz, Gödel, product (Goguen)

## First-Order Logic Example

Logic Review

## A typical first-order logic axiom (formula)

$$\forall x \left( \exists y \left( R(x,y) \land A(y) \right) \right)$$

## Components

variables: x, y

predicates (n-ary relations): R, A

▶ logical operators/connectives:  $\land$ ,  $\lor$ ,  $\neg$ ,  $\rightarrow$ ,  $\leftrightarrow$ 

▶ quantifiers: ∀, ∃

## First-Order Logic Example

#### Introduction

- TN Backgrour
- LTN Evol

### Logic Review

- LTN Component
- LTN Real Log
- LTN Real Logic Axio
- LTN Knowledge Base
- LTN Loss Function
- LTN Training

Github Repo Examples
Binary Classification

LTN Academic Us

Summar

A typical first-order logic axiom (formula)

$$\forall x \left( \exists y \left( R(x,y) \land A(y) \right) \right)$$

### Components

- $\triangleright$  variables: x, y
- ightharpoonup predicates (n-ary relations): R, A
- ▶ logical operators/connectives:  $\land$ ,  $\lor$ ,  $\neg$ ,  $\rightarrow$ ,  $\leftrightarrow$
- ▶ quantifiers: ∀, ∃

## One potential interpretation

Let x and y be people,  $R \equiv$  friend,  $A \equiv$  Italian:

## First-Order Logic Example

#### Introduction

- TN Backgroun
- LTN VVILIIII INS

#### Logic Review

- LTN.C
- ETTA Componer
- LTN Real Logic
- LTN Knowledge Base
- LTN Loss Function
- LTN Training
- LIN Training
- Github Repo Examples
  Binary Classification
- LTN Academic Us

Summa

A typical first-order logic axiom (formula)

$$\forall x \left( \exists y \left( R(x,y) \land A(y) \right) \right)$$

### Components

- $\triangleright$  variables: x, y
- ▶ predicates (*n*-ary relations): *R*, *A*
- ▶ logical operators/connectives:  $\land$ ,  $\lor$ ,  $\neg$ ,  $\rightarrow$ ,  $\leftrightarrow$
- ▶ quantifiers: ∀, ∃

### One potential interpretation

Let x and y be people,  $R \equiv$  friend,  $A \equiv$  Italian:

axiom translation: "everyone has a friend that's Italian"

#### Introduction

LTN Backgroun

LTN Within NS

Lauta Basil

LTN Component

#### LTN Real Logic

LTN Real Logic Axio

LTN Loss Funct

LTN Trainin

LTN Toy Example

Binary Classification

LTN Academic Us

Summar

## What is LTN Real Logic?

a fuzzy, first-order logic language for NSI

#### Introduction

LTN Backgroun

LIN Within NS

Logic Povi

LTN Componen

#### LTN Real Logic

LTN Real Logic Axion LTN Knowledge Base

LTN Training

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Us

Summar

## What is LTN Real Logic?

▶ a fuzzy, first-order logic language for NSI

## Components of the language

individuals (constants) are real-valued tensors

#### Introduction

LTN Backgroun

LTN Within NS

Logic Revie

LTN Component

LTN Real Logic

ITN D II

LTN Knowledge Base

LTN Loss Fun

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Us

Summar

## What is LTN Real Logic?

► a fuzzy, first-order logic language for NSI

- ▶ individuals (constants) are real-valued tensors
- variables are collections of individuals

#### Introduction

LTN Backgroun

LTN Within NSI

Logic Revi

LTN Componen

#### LTN Real Logic

LTN Real Logic Axio LTN Knowledge Base

LTN Loss Fun

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

S...mman

## What is LTN Real Logic?

► a fuzzy, first-order logic language for NSI

- ▶ individuals (constants) are real-valued tensors
- variables are collections of individuals
- predicates are:
  - neural networks that output truth degrees  $\in [0,1]$
  - functions that output truth degrees  $\in [0,1]$

#### Introduction

LTN Backgroun

LTN Within NSI

Logic Povi

LTN Componen

ETTA Componen

#### LTN Real Logic

LTN Knowledge Base

LTN Loss Fun

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Us

Summarv

## What is LTN Real Logic?

▶ a fuzzy, first-order logic language for NSI

- ▶ individuals (constants) are real-valued tensors
- variables are collections of individuals
- predicates are:
  - neural networks that output truth degrees  $\in [0,1]$
  - functions that output truth degrees  $\in [0,1]$
- operators have (differentiable) fuzzy logic semantics

#### Introduction

- LTN Backgroun
- LTN Within NS
- Logic Revie
- LTN Componen
- ETTA Componer

#### LTN Real Logic

- LTN Keal Logic Axio
- LTN Loss Function
- LTN Training
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Us

Summary

## What is LTN Real Logic?

▶ a fuzzy, first-order logic language for NSI

- ▶ individuals (constants) are real-valued tensors
- variables are collections of individuals
- predicates are:
  - neural networks that output truth degrees  $\in [0,1]$
  - functions that output truth degrees  $\in [0,1]$
- operators have (differentiable) fuzzy logic semantics
- quantifiers have (differentiable) fuzzy logic semantics

#### Introduction

LTN Backgroun

LTN Evolution

Logic Revie

LTN Component

#### LTN Real Logic

LTN Real Logic Axio

LTN Loss Function

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Us

Summar

## Various fuzzy **semantics** exist for operators/quantifiers

- ▶ Łukasiewicz, Gödel, product (Goguen), others ...
- ► LTN API offers a wide selection of these semantics

#### Introduction

TN Backgroun

LTN Evolut

ITN C

#### LTN Real Logic

LTN Real Logic Axio LTN Knowledge Base

LTN Loss Func

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Us

Summar

## Various fuzzy **semantics** exist for operators/quantifiers

- Lukasiewicz, Gödel, product (Goguen), others ...
- ► LTN API offers a wide selection of these semantics

### But: LTN recommends a specific subset of semantics

- called the "stable product configuration"
- it's the safe choice semantics for each operator/quantifier

#### LTN Real Logic

### Various fuzzy **semantics** exist for operators/quantifiers

- - Lukasiewicz, Gödel, product (Goguen), others ...
  - ▶ LTN API offers a wide selection of these semantics

## But: LTN recommends a specific **subset** of semantics

- called the "stable product configuration"
- it's the **safe choice semantics** for each operator/quantifier

Why?

#### Introduction

LTN Backgroun LTN Within NSI

LITTE EVOIDE

LTN Componen

LTN Real Logic

LTN Real Logic Axio

LTN Loss Fun

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic U

Summar

## Various fuzzy **semantics** exist for operators/quantifiers

- Lukasiewicz, Gödel, product (Goguen), others ...
- ▶ LTN API offers a wide selection of these semantics

## But: LTN recommends a specific subset of semantics

- called the "stable product configuration"
- it's the safe choice semantics for each operator/quantifier

## Why?

 alternative semantics induce vanishing or exploding or single-passing gradients

#### LTN Real Logic

### Various fuzzy **semantics** exist for operators/quantifiers

- Łukasiewicz, Gödel, product (Goguen), others ...
- ▶ LTN API offers a wide selection of these semantics

## But: LTN recommends a specific **subset** of semantics

- called the "stable product configuration"
- it's the **safe choice semantics** for each operator/quantifier

## Why?

- alternative semantics induce vanishing or exploding or single-passing gradients
- source: van Krieken et al. (2020). "Analyzing Differentiable Fuzzy Logic Operators". arXiv:2002.06100

#### Introduction

TN Backgroun

LTN Within NS

Logic Review

LTN Componen

#### LTN Real Logic

LTN Real Logic Axio LTN Knowledge Base LTN Loss Function

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Use

Summary

### **Stable Logical Operator Semantics**

Operator	Semantics *	Fuzzy Logic Source System
And (∧)	$u \wedge v = uv$	product
Or (∨)	$u \vee v = u + v - uv$	product
Not $(\neg)$	$\neg u = 1 - u$	Łukasiewicz
Implies $( o)$	$u \to v = 1 - u + uv$	Reichenbach **
Equiv $(\leftrightarrow)$	$u \leftrightarrow v = (u \to v) \land (v \to u)$	(mixed)

<sup>\*</sup> u and v represent degrees of truth  $\in [0,1]$ 

<sup>\*\*</sup>  $u \to v \equiv (\neg u) \lor v = (1-u) \lor v = (1-u) + v - (1-u)v = 1 - u + uv$ 

**Existential quantifier** ∃ - 'there exists' (at least one)

### Stable Semantics

The "generalised mean" (p-mean), here denoted pM:

$$pM(u_1, \dots, u_n) = \left(\frac{1}{n} \sum_{i=1}^n u_i^p\right)^{\frac{1}{p}} \quad p \ge 1$$

LTN Real Logic

**Existential quantifier** ∃ - 'there exists' (at least one)

### Stable Semantics

The "generalised mean" (p-mean), here denoted pM:

$$pM(u_1, \dots, u_n) = \left(\frac{1}{n} \sum_{i=1}^n u_i^p\right)^{\frac{1}{p}} \quad p \ge 1$$

where:

- $\mathbf{v}_i \in [0,1] \quad i = 1 \dots n \quad \text{(degrees of truth)}$
- ightharpoonup 0 < pM < 1
- ightharpoonup if p=1, pM= arithmetic mean
- ightharpoonup as  $p \to \infty$ ,  $pM \to max(u_1, \ldots, u_n)$

#### LTN Real Logic

Introduction

TN Background

Lauta Davida

LTN Componen

LTN Real Logic

#### LIN Real Log

LTN Knowledge Ba

LTN Loss Function

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

**Existential quantifier** ∃ - 'there exists' (at least one)

### Stable Semantics

The "generalised mean" (p-mean), here denoted pM:

$$pM(u_1, \dots, u_n) = \left(\frac{1}{n} \sum_{i=1}^n u_i^p\right)^{\frac{1}{p}} \quad p \ge 1$$

#### where:

- ightharpoonup 0 < pM < 1
- ightharpoonup if p=1, pM= arithmetic mean
- ightharpoonup as  $p \to \infty$ ,  $pM \to max(u_1, \ldots, u_n)$

So, as  $p \to \infty$ , pM approximates a maximum function

Universal quantifier  $\forall$  - 'for all'

### Stable Semantics

The "generalised mean of deviations from the truth" (p-mean error), here denoted pME:

$$pME(u_1, ..., u_n) = 1 - \left(\frac{1}{n} \sum_{i=1}^n (1 - u_i)^p\right)^{\frac{1}{p}} \quad p \ge 1$$

Introduction

LTN Background

LTN Within N: LTN Evolution

Logic Review

LTN Component

LTN Real Logic

LTN Real Logic Axio

LTN Loss Function

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Us

Summar

# Real Logic "stable product configuration"

**Universal quantifier** ∀ - 'for all'

### Stable Semantics

The "generalised mean of deviations from the truth" (p-mean error), here denoted pME:

$$pME(u_1, ..., u_n) = 1 - \left(\frac{1}{n} \sum_{i=1}^n (1 - u_i)^p\right)^{\frac{1}{p}} \quad p \ge 1$$

where:

- $ightharpoonup 0 \le pME \le 1$
- ▶ if p = 1, pME = arithmetic mean
- ightharpoonup as  $p \to \infty$ ,  $pME \to min(u_1, \dots, u_n)$

#### Introduction

LTN Background

LTN Within NSI

Logic Povic

LTN Componen

LTN Real Logic

LTN Knowledge Base

LTN Loss Fur

Cithub Repo Examples
Binary Classification

LTN Academic Us

▶ a

# Real Logic "stable product configuration"

### **Universal quantifier** $\forall$ - 'for all'

### Stable Semantics

The "generalised mean of deviations from the truth" (p-mean error), here denoted pME:

$$pME(u_1, \dots, u_n) = 1 - \left(\frac{1}{n} \sum_{i=1}^n (1 - u_i)^p\right)^{\frac{1}{p}} \quad p \ge 1$$

#### where:

- ightharpoonup 0 < pME < 1
- ightharpoonup if p=1, pME= arithmetic mean
- ightharpoonup as  $p \to \infty$ ,  $pME \to min(u_1, \ldots, u_n)$

So, as  $p \to \infty$ , pME approximates a minimum function

#### LTN Real Logic

Introduction to the Logic Tensor Networks (LTN) Framework

# Real Logic "stable product configuration"

Introduction

l Backgroun

LTN Evolution

Logic Revie

LTN Component

#### LTN Real Logic

LTN Real Logic Axion LTN Knowledge Base LTN Loss Function

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clastering

LTN Academic Us

Summary

### The LTN "stable product configuration" in code

quantifier p-values are **hyper-parameters** to be tuned

Introduction

LTN Backgroun

TIV Dackgrou

LTN Evo

Lauta Basila

LTN Componer

LTN Real Logi

LTN Real Logic Axioms

LTN Knowledge Bar

LTN Knowledge Base

LTN Loss Function

LTN Training

Cithub Repo Example

Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

Summar

What is an LTN Real Logic axiom?

Introduction

LTN Backgrour

LTN Within NS

. . . . .

LING

LTN Component

LTN Real Logic Axioms

LTN Knowledge Base

LTN Lass Evention

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Us

Summan

### What is an LTN Real Logic axiom?

▶ an assertion of knowledge over data

#### Introduction

LTN Backgrour

LTN Evolution

Logic Review

LTN Componen

LTN Real Logic

LTN Real Logic Axioms

LTN Knowledge Base

TN Loss Function

LTN Trainin

LTN Toy Example Github Repo Example Binary Classification

LTN Academic Us

Summan

### What is an LTN Real Logic axiom?

- an assertion of knowledge over data
- a logical constraint to be satisfied

#### Introduction

- TN Backgroun
- LTN Evolution
- Logic Revie
- LTN Componer
- LTN Real Logic
- LTN Real Logic Axioms
- LTN Knowledge Base
- LTN Loss Fund
- LTN Training
- Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Us

Summar

### What is an LTN Real Logic axiom?

- ► an assertion of knowledge over data
- a logical constraint to be satisfied
- executable Python code
- $\blacktriangleright$  a nested function call that  ${\bf returns}$  a  ${\bf satisfaction}$  level (truth degree)  $\in [0,1]$

Introduction

TN Backgroun

LTN Within NS

Logic Revie

LTN Componer

LTN Real Logic

LTN Real Logic Axioms

LTN Loss Function

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

Summary

### What is an LTN Real Logic axiom?

- ► an assertion of knowledge over data
- ▶ a logical constraint to be satisfied
- executable Python code
- $\blacktriangleright$  a nested function call that returns a satisfaction level (truth degree)  $\in [0,1]$

Recall our example first-order logic axiom

$$\forall x \left( \exists y \left( R(x,y) \wedge A(y) \right) \right)$$

#### Introduction

- TN Backgroun
- LTN Within N
- Logic Revie
- LTN Componer
- LTN Real Logic

#### LTN Real Logic Axioms

- LTN Knowledge Ba
- LTN Training
- LTN Toy Examples Github Repo Examples Binary Classification
- LTN Academic Us

Summan

### What is an LTN Real Logic axiom?

- ► an assertion of knowledge over data
- ▶ a logical constraint to be satisfied
- executable Python code
- $\blacktriangleright$  a nested function call that returns a satisfaction level (truth degree)  $\in [0,1]$

### Recall our example first-order logic axiom

$$\forall x \left( \exists y \left( R(x,y) \wedge A(y) \right) \right)$$

### The equivalent LTN Real Logic axiom might look like

 $sat_level = Forall(x, Exists(y, And(Friend(x,y), Italian(y))))$ 

Introduction

LTN Backgroun

LTN Evolution

Lauta Basila

LTN Component

LTN Real Logic

LTN Real Logic Axioms

LTN Knowledge Race

TN Loss Eunstion

LTN Training

LTN Toy Example

Github Repo Examples Binary Classification Clustering

LTN Academic Us

Summar

### Our example axiom again

$$\forall x \left( \exists y \left( R(x,y) \wedge A(y) \right) \right)$$

=

 $Forall(x, \ Exists(y, \ And(Friend(x,y), \ Italian(y))))$ 

Introduction

LTN Backgrou

LIN Within N

Logic Revie

LTN Component

Live Componer

LTN Real Logic Axioms

LTN Knowledge Base

LTN Loss Function

LTN Training

LTN Toy Example

Github Repo Examples
Binary Classification
Clustering

LTN Academic Us

Summar

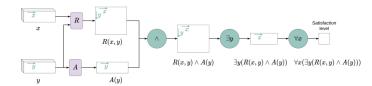
### Our example axiom again

$$\forall x \left( \exists y \left( R(x,y) \wedge A(y) \right) \right)$$

 $\equiv$ 

Forall(x, Exists(y, And(Friend(x,y), Italian(y))))

### Computational graph for our example axiom



Introduction

LTN Backgroun

LTN Codes

Lauta Basta

LTN Component

ETTA Componen

LTN Real Logi

ETTA TICUI LOGIC 7 IXII

LTN Knowledge Base

LTN Loss Func

LTN Training

Github Repo Example

Binary Classification

LTN Academic Us

Summar

What is an LTN Knowledge Base (KB)?

#### Introduction

LTN Backgroun

LTN Evol

Logic Review

LTN Componen

LTN Real Logic

LTN Knowledge Base

LTN Knowledge base

LTN Training

LTN Toy Example

Github Repo Example
Binary Classification
Clustering

LTN Academic Use

Summar

### What is an LTN Knowledge Base (KB)?

A collection of LTN Real Logic axioms over data

implemented within a Python function

#### Introduction

LTN Backgroun

LTN Evol

Logic Revie

LTN Componer

LTN Real Logic

LTN Knowledge Base

LTN Loss Funct

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Us

Summar

### What is an LTN Knowledge Base (KB)?

A collection of LTN Real Logic axioms over data

- implemented within a Python function
- $\blacktriangleright$  that **returns** a single, overall KB **satisfaction level** (truth degree)  $\in [0,1]$

#### Introductio

LTN Backgroun

Logic rierien

LTN Componer

LTN Real Logic

LTN Knowledge Base

LTN Loss Fund

LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Use

Summar

### What is an LTN Knowledge Base (KB)?

A collection of LTN Real Logic axioms over data

- ▶ implemented within a **Python function**
- $\blacktriangleright$  that returns a single, overall KB satisfaction level (truth degree)  $\in [0,1]$

### KB 'formula aggregator'

- **aggregates** the **satisfaction levels** of individual axioms
- computes the overall KB satisfaction level

#### Introductio

- TN Backgroun
- LTN Evolu
- Logic Revie
- LTN Componer
- LTN Real Logic
- LTN Real Logic
- LTN Knowledge Base
- LTN Loss Fun
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Use

Summa

### What is an LTN Knowledge Base (KB)?

A collection of LTN Real Logic axioms over data

- ▶ implemented within a **Python function**
- $\blacktriangleright$  that returns a single, overall KB satisfaction level (truth degree)  $\in [0,1]$

### KB 'formula aggregator'

- aggregates the satisfaction levels of individual axioms
- computes the overall KB satisfaction level
- ▶ has same semantics as the universal quantifier (∀)
  - the *p*-mean error (pME), typically with p=2
  - a (lenient) approximator of the minimum satisfaction level

# LTN Knowledge Base – toy examples

Introduction

TN Backgroun

LTN Evolu

. . . . .

LTN Componen

LTN D. II

LTN Real Logic Axio

LTN Knowledge Base

LIN Loss Funct

LTN Training

Github Repo Example Binary Classification Clustering

LTN Academic Us

Summan

### Toy example KB with 1 axiom

```
def axioms(x,y):
    sat_level = Forall(x, Exists(y, And(Friend(x,y), Italian(y))))
    return sat_level
```

# LTN Knowledge Base – toy examples

#### Introductio

- LTN Background
- LTN Color
- . . . . .

#### --8-- -----

- LTN Componen
- LTN Real Logic
- LTN Real Logic Axion
- LTN Knowledge Base
- LTN Loss Function
- LTN Toy Examples
  Github Repo Examples
- LTN Academic He

Summar

### Toy example KB with 1 axiom

```
def axioms(x,y):
    sat_level = Forall(x, Exists(y, And(Friend(x,y), Italian(y))))
    return sat_level
```

### Toy example KB with 2 axioms

# The (standard) LTN Loss Function

Introduction

LTN Backgroun

LTN Contract

. . . . . .

LTM Component

LIN Componen

LTN Real Logic Axio

LTN Knowledge Base

LTN Loss Function

LTN Training

LTN Toy Exampl

Github Repo Example
Binary Classification

LTN Academic Us

Summar

We call the KB and it returns

 $sat\_level \in [0,1]$ 

# The (standard) LTN Loss Function

Introduction

LTN Backgroun

LTN Evolution

Logic Revi

LTN Component

LTN Componen

LTN Real Logic Axio

LTN Knowledge Base

#### LTN Loss Function

LTN Training

LTN Toy Examples

Binary Classification
Clustering

LTN Academic Us

Summa

We call the KB and it returns

 $sat\_level \in [0,1]$ 

The (standard) LTN loss function is simply

$$loss = 1 - sat\_level$$

As  $sat\_level \rightarrow 1$ ,  $loss \rightarrow 0$ .

Introduction to the Logic Tensor Networks (LTN) Framework

# A representative LTN training loop

```
ds train = # conventional training set
model = MyNeuralNetworkModelClass()
Pred = ltn.Predicate(model) # NN (predicate) used in KB axioms
trainable vars = Pred.trainable variables
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
for epoch in range(n_epochs):
    for features, labels in ds train:
        with tf.GradientTape() as tape:
            sat level = axioms(features, labels)
            loss = 1 - sat level
        grads = tape.gradient(loss, trainable vars)
        optimizer apply gradients(zip(grads, trainable vars))
```

#### LTN Training

#### How the LTN framework trains neural networks:

- gradient descent drives loss minimisation
- loss minimised by maximising KB satisfiability
- KB satisfiability maximised by the NN learning the function that best satisfies all the axioms

LTN Backgroup

I TNI Within NSI

LTN Evo

Logic Review

LTN Component

LTN Real Los

LTN Real Logic Axid

LTN Knowledge B

LTN Loss Funct

LIN Iraining

LTN Toy Examples

Github Repo Examples

Clustering

LTN Academic Us

Summary

# Github Examples Illustrating Range of LTN

Note: The Github examples illustrate the mechanisms and flexibility of LTN, not how it's intended to be used in practice.

.TN Backgroui

LTN Within N

Logic Revie

LTN Component

LTN D. II.

LTN Real Logic Axid

LTN Knowledge Bas

LTN Loss Funct

LTN Training

LTN Toy Examples

Github Repo Examples

Clustering

LTN Academic Us

Summar

# Github Examples Illustrating Range of LTN

Note: The Github examples illustrate the mechanisms and flexibility of LTN, not how it's intended to be used in practice.

Classification & Regression (supervised)

LTN Backgrour

LTN Within N

Logic Review

LTN Component

LTN D. LL.

LTN Real Logic Axio

LTN Knowledge Base LTN Loss Function

LTN Trainin

LTN Toy Examples

Github Repo Examples

Clustering

LTN Academic Us

Summar

# Github Examples Illustrating Range of LTN

Note: The Github examples illustrate the mechanisms and flexibility of LTN, not how it's intended to be used in practice.

Classification & Regression (supervised)

Clustering (unsupervised)

LTN Backgrou

LTN Within NSI

LTN Evolution

. ....

LTN Component

LTN Real Log

LTN Knowledge Bas

LTN Loss Function

LTN Train

LTN Tov Examples

Github Repo Examples

lustering

LTN Academic Us

Summar

### Github Examples Illustrating Range of LTN

Note: The Github examples illustrate the mechanisms and flexibility of LTN, not how it's intended to be used in practice.

Classification & Regression (supervised)

Clustering (unsupervised)

Addition with MNIST digits (semi-supervised)

- LTN Backgrou
- LTN Within NSI
- Lauta Basil
- . . . . .
- LTN Componen
- LTN Real Log
- LTIV Real Logic Axio
- LTN Knowledge Basi
- LIN Loss Funct
- LIN Irainii
- LTN Toy Examples

### Github Repo Examples

Clustering

LTN Academic Us

Summar

# Github Examples Illustrating Range of LTN

Note: The Github examples illustrate the mechanisms and flexibility of LTN, not how it's intended to be used in practice.

Classification & Regression (supervised)

Clustering (unsupervised)

Addition with MNIST digits (semi-supervised)

Learning embeddings for individuals

### (LTN) Framework

Introduction

LTN Backgro

LTN Evol

Logic Review

LTN Componer

LTN Real Lo

LTN Knowledge Ras

LTN Loss Func

LTN Training

LTN Toy Examples

Github Repo Examples

nary Classific ustering

LTN Academic Us

Summar

### Github Examples Illustrating Range of LTN

Note: The Github examples illustrate the mechanisms and flexibility of LTN, not how it's intended to be used in practice.

Classification & Regression (supervised)

Clustering (unsupervised)

Addition with MNIST digits (semi-supervised)

Learning embeddings for individuals

### Deductive reasoning

- determining whether a proposition is entailed by a KB?
- ightharpoonup e.g.  $(A \lor B) \models A$  ?
- reasoning by refutation: learning a counter-example where  $(A \lor B)$  is satisfied but A is not satisfied

Introduction

LTN Backgroun

LTN MOLE NO

LTN Evo

Logic Review

LTN Component

LTM Componen

LTN Real Logic

LTN Knowledge Bac

LTN Loss Function

LTN Training

LTN Toy Examples

Binary Classification

Clustering

LTN Academic Us

Introduction

LTN Backgroun

LTN F

LTN Component

LTM Componen

LTN Real Logic Axio

LTN Knowledge Base

LTN Loss Functi

LTN Training

Github Repo Examples

Binary Classification

Clustering

LTN Academic Us

Summar

Predicate (NN): **Cat** – cat classifier

Introduction

LTN Backgroun

LTN Fredrice

Logic Review

LTN Componen

Live Compone

LTN Real Log

LTN Knowledge Base

LTN Loss Funct

LTN Training

Github Repo Examples

Binary Classification

LTN Academic Us

Summary

### Predicate (NN): **Cat** – cat classifier

```
def axioms(features, labels):
    pos = ltn.Variable("pos", features[labels])
    neg = ltn.Variable("neg", features[tf.logical_not(labels)])
    axioms = [
        Forall(pos, Cat(pos)),
        Forall(neg, Not(Cat(neg)))
    ]
    sat_level = formula_aggregator(axioms)
    return sat_level
```

Introduction

LTN Backgroun

LIN Within NSI

Logic Revie

LTN.C

LTN Componer

LTN Real Log

LTN Knowledge Base

LTN Loss Function

LTN Training

Github Repo Example

Binary Classification Clustering

LTN Academic Us

Summar

Predicate (NN): **Cat** – cat classifier

### Observe:

here the **knowledge** is limited to **what's in the data** 

N Backgroui

LTN Evolution

Logic Revie

LTN Componen

LITTE COMP

LTN Real Logic

LTN Knowledge Bas

LTN Loss Functi

LTN Training

Github Repo Example Binary Classification

Clustering

LIN Academic Us

Summar

Predicate (NN): **Cat** – cat classifier

### Observe:

- here the **knowledge** is limited to **what's in the data**
- label supervision replaced by constraints to be satisfied

# An LTN KB for Clustering

Introduction

LTN Backgroun

...........

LTN Evol

Logic Review

LTN Component

Live Component

LTN Real Logic

LTN Knowledge Bac

LTN L --- E----i--

LTN Training

LTN Toy Example

Binary Classification

Clustering

LTN Academic Us

# An LTN KB for Clustering

Predicate (NN): C - cluster membership classifier

### LTN Darkenson

LTN Background

LTN Evol

Logic Povio

#### LTN Component

LTN Real Los

LTN Real Logic Axio

LTN Knowledge Base

LTN Loss Functi

LTN Training

Github Repo Example

#### Clustering

LTN Academic Us

# An LTN KB for Clustering

### Predicate (NN): **C** – cluster membership classifier

```
cluster = ltn.Variable("cluster".clst ids) # 0...K-1
# Axiom interpretations:
# 1: every example should be assigned to a cluster
# 2: every cluster should be non-empty
# 3: if the points are near, they should belong to the same cluster
# 4: if the points are far, they should belong to different clusters
# 5: clusters should be disjoint: handled IMPLICITLY by the softmax
    activation function in the output layer of predicate C. which
    returns probabilities that are mutually exclusive
def axioms(x,y):
   axioms = [
        Forall(x, Exists(cluster, C([x,cluster]))),
        Forall(cluster, Exists(x, C([x,cluster]))),
        Forall([cluster,x,y], Equiv(C([x,cluster]),C([y,cluster])),
            mask = is less than([eucl dist([x,v]).close thr])).
        Forall([cluster,x,y], Not(And(C([x,cluster]),C([y,cluster]))),
            mask = is greater than([eucl dist([x,v]),distant thr]))
    sat level = formula aggregator(axioms)
    return sat_level
```

Introduction

LTN Background

LTN Evolution

LTN Components

LTN Real Logic

LTN Real Logic Axiom LTN Knowledge Base

LTN Loss Funct

Github Repo Example

Clustering

LTN Academic Us

Introduction to the Logic Tensor Networks (LTN) Framework

#### Introduct

LTN Background

LTN Within NSI

Logic Revie

LTN Componen

LTN Componer

LTN Real Logic Axio

LTN Loss Function

LTN Training

Github Repo Examples
Binary Classification

#### Clustering

LTN Academic Us

Summary

# An LTN KB for Clustering

### Predicate (NN): **C** – cluster membership classifier

```
cluster = ltn.Variable("cluster".clst ids) # 0...K-1
# Axiom interpretations:
# 1: every example should be assigned to a cluster
# 2: every cluster should be non-empty
# 3: if the points are near, they should belong to the same cluster
# 4: if the points are far, they should belong to different clusters
# 5: clusters should be disjoint: handled IMPLICITLY by the softmax
    activation function in the output layer of predicate C. which
    returns probabilities that are mutually exclusive
def axioms(x,y):
   axioms = [
        Forall(x, Exists(cluster, C([x,cluster]))),
        Forall(cluster, Exists(x, C([x,cluster]))),
        Forall([cluster,x,y], Equiv(C([x,cluster]),C([y,cluster])),
            mask = is_less_than([eucl_dist([x,y]),close_thr])),
        Forall([cluster,x,y], Not(And(C([x,cluster]),C([y,cluster]))),
            mask = is greater than([eucl dist([x,v]),distant thr]))
    sat level = formula aggregator(axioms)
    return sat_level
```

### Observe:

▶ here the **knowledge** is all **external** (supplementary) to the **data** 

# Research Work using LTN

Introduction

LTN Backgroun

LTN Evol

Lauta Davida

LTN Component

LIN Component

LTN Real Log

LTIN Real Logic Axi

LTTV Trilowieuge Da

LIN LOSS FUNC

LTN Training

Github Repo Example

Binary Classification Clustering

LTN Academic Use

Summar

We briefly review 4 research papers using LTN

### Introductio

LTN Background

ogic Revie

LTN Component

LTN Real Log

LTN Real Logic Axis

LTN Knowledge Base

LIN LOSS FUN

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification

LTN Academic Use

Summan

## Paper

Donadello & Serafini (2019). Compensating Supervision Incompleteness with Prior Knowledge in Semantic Image Interpretation.

### Introduction

- LTN Backgro
- LTN W

### Logic Revi

- LTN Componer
- LTN Real Lo
- LTN Real Logic Axio
- LTN Knowledge Bas
- LTN Loss Fun
- LTN Training

LTN Toy Examples Github Repo Examples Binary Classification Clustering

LTN Academic Use

Summar

## Paper

Donadello & Serafini (2019). Compensating Supervision Incompleteness with Prior Knowledge in Semantic Image Interpretation.

## Description

- task: predict (subject,relation,object) triples (visual relationships) in images; e.g. (person,ride,horse)
- approach:
  - object detector detects and classifies subject/object bboxes
  - LTN-trained NNs predict the relations
- strategy: inject relational knowledge in the form of negative domain and range constraints
  - e.g.  $\forall xy \; Ride(x,y) \rightarrow Not(Table(x))$
  - e.g.  $\forall xy \; Ride(x,y) \rightarrow Not(Chair(y))$

nb: the unary predicates (e.g. Table(x)) for the object classes are rule-based functions returning 1 or 0; the binary predicates (e.g. Ride(x,y)) for the relations are trainable NNs

### Introduction

LTN Backgroun

LTN Evolution

Logic Review

LTN Componen

LTN Real Log

LTN Real Logic Axio

LTN Knowledge Base

TN T-:-:-

LTN Training

Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

Summan

## Results: NNs trained over LTN prior knowledge ...

- deliver better relation predictive performance
- including better zero-shot relation predictions

### Introduction

- LTN Backgroun
- LTN Within NSI LTN Evolution
- Logic Review
- LTN Componen
- LTN Real Logic
- LTN Real Logic A
- LTN Knowledge Base
- LTN Loss Function
- LTN Training
- LTN Toy Examples Github Repo Examples Binary Classification Clustering
- LTN Academic Use

Summar

## Results: NNs trained over LTN prior knowledge ...

- deliver better relation predictive performance
- ▶ including better **zero-shot** relation predictions

## Issue: LTN scalability limitation

- dataset has 70 relations and 100 object classes
- $\triangleright$   $\approx$  25 relations permit domain/range restrictions
  - $\Rightarrow \approx$  **25 NNs** to be trained
- ▶ 100 domain + 100 range constraints  $\approx$  200 axioms/relation
  - $\Rightarrow$  25 relations  $\times$  200 axioms  $\approx$  **5000 axioms** in KB
- **repeated calls** to the same NNs per mini-batch
  - $\Rightarrow$  pprox 200 calls per binary predicate NN
- ▶ (nb: Donadello uses only a "tractable sample")

# LTN with Deep Reinforcement Learning

### Introduction

- LTN Backgroui
- LTN Within NS
- Logic Revie
- LTN Compone
- Little Compo
- LTND II
- LTN Knowledge Race
- LTN I F ...
- LIIV LOSS I UI
- LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

Summary

## Paper

Badreddine & Spranger (2019). Injecting Prior Knowledge for Transfer Learning into Reinforcement Learning Algorithms using Logic Tensor Networks.

### Discussion

- ► task: grid game; agent needs to learn to collect certain objects and avoid others to maximise cumulative reward
- ► LTN used to **preprocess** input data to **derive explicit knowledge** of which grid cells to target/avoid
- ▶ they show agent learns to exploit the prior knowledge
- role of LTN here is minor (incidental)

## (LTN) Framework

### Introduction

TN Background

LTN Within N

Logic Revi

### LTN Component

LTN Real Log

LTN Real Logic A

LTN Knowledge Bas

LTN Loss Func

LTN Training

Cithub Repo Examples

Binary Classification Clustering

LTN Academic Use

Summary

# LTN with Deep Reinforcement Learning

## Knowledge derivation; CNN architecture for RL system

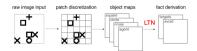


Figure 5: Priors on the grid game

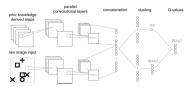


Figure 6: A Double Dueling architecture architecture with conjoint image and prior knowledge inputs.

## LTN axioms for one preprocessing scenario

```
\forall x \in \mathcal{O} : \operatorname{circle}(x) \leftrightarrow \operatorname{goto}(x)
\forall x \in \mathcal{O} : \operatorname{cross}(x) \leftrightarrow \operatorname{avoid}(x)
```

### Introduction

- LTN Backgrou
- LIN W
- Logic Revi
- LTN Componer
- ....
- LTN Knowledge Ras
- LTN Loss Function
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering

### LTN Academic Use

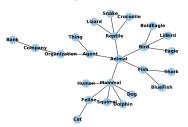
Summary

## Paper

► Bianchi & Hitzler (2019). On the Capabilities of Logic Tensor Networks for Deductive Reasoning.

## One experiment: taxonomy reasoning

▶ train set: 24-class taxonomy with 23 subClassOf relations



- ▶ test set: the full transitive closure + all false relations
- ightharpoonup objective: train NN to predict sub(x,y) given arbitrary x,y

### Introductio

TN Backgrour

LTN Evolu

Logic Review

LTN Component

ETTA COMPONE

LTTV NCUI LOGI

LTN Knowledge Rase

LTN Loss Function

LTN Trainin

LTN Toy Examples Github Repo Examples Binary Classification

LTN Academic Use

Summar

### LTN KB axioms

- $ightharpoonup orall a,b,c \quad sub(a,b) \wedge sub(b,c) 
  ightarrow sub(a,c)$  (transitive)
  - $ightharpoonup \forall a \qquad \neg sub(a,a)$  (not reflexive)
- $\blacktriangleright \ \, \forall a,b \qquad sub(a,b) \to \neg sub(b,a) \qquad \qquad \text{(not symmetric)}$

### Introduction

- TN Backgrour
- LTN Within NS
- Logic Review
- LTN Componen
- LTN Real Logic
- LTN Real Logic Ax
- LTN Knowledge Base
- LTN Loss Function
- LTN Training
- Github Repo Examples
  Binary Classification
  Clustering

### LTN Academic Use

Summar

### LTN KB axioms

- $ightharpoonup orall a, b, c \quad sub(a,b) \wedge sub(b,c) 
  ightarrow sub(a,c)$  (transitive)
- $ightharpoonup \forall a \qquad \neg sub(a,a)$  (not reflexive)
- $\blacktriangleright \ \forall a,b \qquad sub(a,b) \to \neg sub(b,a) \qquad \qquad \text{(not symmetric)}$

# Results: best performing model (highest satisfiability)

$Pred \setminus Actual$	T	F	Precision
Т	55 (tp)	36 (fp)	0.60
F	26 (fn)	459 (tn)	
Recall	0.68		$576 = 24 \times 24$

- ▶ accuracy of 0.89
  - but a naive classifier predicting only 0s delivers 0.84

### Introduction

LTN Backgrou

LTN Within NSI

Logic Revie

LTN Component

LTN Real Log

LTN Real Logic Axio

LTN Knowledge Base

LIN LOSS Func

LTN Training

Github Repo Example

Binary Classification Clustering

LTN Academic Use

Summar

## LTN general observations:

- ▶ higher KB **satisfiability** ⇒ better predictive **performance**
- ightharpoonup more **axioms** in KB  $\Rightarrow$  better predictive **performance**

### Introduction

- TN Backgrou
- LTN Fv
- Logic Review
- LTN Componen
- LTN Real Logic
- LTN Real Logic Ax
- LTN Knowledge Bas
- LIN LOSS Fund
- LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

Summar

## LTN general observations:

- ightharpoonup higher KB satisfiability  $\Rightarrow$  better predictive performance
- ▶ more axioms in KB ⇒ better predictive performance

### LTN weaknesses:

- prone to predicting false positives
- **struggle** with **multi-hop** inferences
  - e.g. sub(cat, animal) is a bridge too far
  - (i.e. training NNs to generalise remains a challenge)

### Introduction

- TN Backgrou
- LTN W
- Logic Revi
- LTN Compone
- LTN Real Logic
- LTN Real Logic Axio
- LTN Knowledge Base
- LTN Loss Funct
- LTN Training

Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

Summary

## LTN general observations:

- ▶ higher KB **satisfiability** ⇒ better predictive **performance**
- ▶ more axioms in KB ⇒ better predictive performance

### LTN weaknesses:

- prone to predicting false positives
- **struggle** with **multi-hop** inferences
  - e.g. sub(cat, animal) is a bridge too far
  - (i.e. training NNs to generalise remains a challenge)

## LTN strength: explainability / interpretability

- ► LTN can provide partial explanations
- existing (or new) axiom satisfaction levels can be queried after (or during) training to provide insight into 1) what a model is learning, 2) a model's predictive performance
  - query axiom ≡ execute axiom

## LTN for Fairness

### Introduction

- LTN Backgrou
- LTN Eve

### ogic Revie

- LTN Componer
- LTN Real Logi
- LTN Real Logic
- LTN Knowledge Base
- LTN Loss Function
- LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Use

Summar

## Paper

 Wagner & Garcez (2021). Neural-Symbolic Integration for Fairness in AI.

## One experiment: fairness in credit risk assessment

- objective: train binary classifier to avoid gender bias
  - predicate D denotes **Default** (on granted credit)
- strategy: add gender fairness axioms to LTN KB

	$\forall x \in \mathcal{T}_D$ :		D(x)
	$\forall x \in \mathcal{T}_N$ :		$\neg D(x)$
people split into 5 credit quality groups	$\forall x \in \mathcal{R}_{F1}, y \in \mathcal{R}_{M1}$ :	fairness constraints guide the NN to not learn gender bias	$D(x) \leftrightarrow D(y)$
	$\forall x \in \mathcal{R}_{F2}, y \in \mathcal{R}_{M2}$ :		$D(x) \leftrightarrow D(y)$
	$\forall x \in \mathcal{R}_{F3}, y \in \mathcal{R}_{M3} :$		$D(x) \leftrightarrow D(y)$
	$\forall x \in \mathcal{R}_{F4}, y \in \mathcal{R}_{M4} :$		$D(x) \leftrightarrow D(y)$
	$\forall x \in \mathcal{R}_{F5}, y \in \mathcal{R}_{M5}:$ female male		$D(x) \leftrightarrow D(y)$

results: the **fairness** axioms **suppress** the gender bias

## LTN Academic Use - Reflections

LTN Academic Use

## Uptake of LTN has been limited, so far

- 2 of the 4 papers are by students of the originators of LTN
- ► TensorFlow's transition from v1 to v2 likely a factor
  - TF 2 released Sept 2019
  - LTN for TF 2 not available until Jan 2021

Introduction

LTN Backgroun

14 Dackgroui

LTN Evol

Logic Review

LTN Component

LTN Real Lo

LTN Real Logic Axio

LTN Knowledge Base

LIN Loss Fun

LTN Training

LTN Toy Exampl

Github Repo Examples Binary Classification

Clustering

LTN Academic Us

Summary

### Introduction

TN Backgroun

LTN Evolution

Logic Revie

LTN Componen

LTN Real Logi

LTN Real Logic Axio

LTN Loss Funct

LTN Training

LTN Toy Examples
Github Repo Examples
Binary Classification
Clustering

LTN Academic Us

Summary

## The LTN framework approach to NSI allows one to ...

- ▶ blend **knowledge** with **data** using **logical constraints** expressed in a **fuzzy**, **first-order logic** language
- weave a measure of overall constraint satisfaction into the loss function

Summary

## The LTN framework approach to NSI allows one to ...

- blend knowledge with data using logical constraints expressed in a fuzzy, first-order logic language
- weave a measure of overall constraint satisfaction into the loss function

## Types of knowledge expressible

- **data** knowledge: knowledge internal to the data
- real-world knowledge: knowledge external to the data

### Introduction

LTN Backgroui

LTN Within NS

. . . . .

LTN Component

LIN Componen

LTN Real Logic Axid

LTN Knowledge Base

LTN Loss Function

LTN Training

Github Repo Examples
Binary Classification

LTN Academic Us

Summary

## LTN is **not** a special **type** of neural network

it's a special way of training conventional NNs

### Introduction

- LTN Backgroui
- LTN Evolution
- Logic Review
- LTN Componen
- LTN Real Logic
- LTN Real Logic Axion
- LTN Knowledge Base
- LTN Loss Funct
- LTN Training

# Cithub Repo Example Binary Classification

LTN Academic Us

Summary

## LTN is **not** a special **type** of neural network

- it's a special way of training conventional NNs
- the NN still learns an optimal function

### Introduction

- TN Backgrour
- LTN Evolution

### Logic Review

- LTN Componen
- LTN Real Logi
- LTN Real Logic Axiom
- LTN L --- E-----
- LTN Today
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Us

### Summary

## LTN is **not** a special **type** of neural network

- it's a special way of training conventional NNs
- ▶ the NN still learns an **optimal function**
- optimality governed by knowledge axioms over data

### Introductio

- LTN Backgrour LTN Within NSI
- LTN Within NSI LTN Evolution
- Logic Review
- LTN Componen
- LTN Real Logic
- LTN Real Logic Axio
- LTN Loss Functi
- LTN Training
- LTN Toy Examples Github Repo Examples
- LTN Academic Use

Summary

## LTN is **not** a special **type** of neural network

- it's a special way of training conventional NNs
- ► the NN still learns an **optimal function**
- optimality governed by knowledge axioms over data
- ▶ the optimal function learned by the NN is the one that best satisfies the knowledge axioms (given the data)

### Introduction

- LTN Backgroun
- LTN Within NSI LTN Evolution
- Logic Review
- LTN Componer
- LTN Real Logic
- LTN Real Logic Axiom
- LTN Loss Functi
- LTN Training
- LTN Toy Examples
  Github Repo Examples
- LTN Academic Us

Summary

## LTN is **not** a special **type** of neural network

- it's a special way of training conventional NNs
- ▶ the NN still learns an optimal function
- optimality governed by knowledge axioms over data
- ▶ the optimal function learned by the NN is the one that best satisfies the knowledge axioms (given the data)

### Miscellaneous observations

► LTN is **flexible** enough to handle **many** ML tasks

### Introduction

- LTN Backgrour LTN Within NSI
- LTN Within NSI LTN Evolution
- Logic Revier
- LTN Componen
- LIN Keal Logic
- ITN Knowledge Bas
- LTN Loss Functi
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Us

Summary

## LTN is **not** a special **type** of neural network

- it's a special way of training conventional NNs
- ▶ the NN still learns an **optimal function**
- ▶ optimality governed by knowledge axioms over data
- ▶ the optimal function learned by the NN is the one that best satisfies the knowledge axioms (given the data)

- ► LTN is **flexible** enough to handle **many** ML tasks
- LTN academic up-take has been limited, so far

### Introduction

- LTN Backgroun LTN Within NSI
- Logic Review
- LTN Componer
- LTN Real Logic
- LTN Knowledge Base
- LTN Loss Functi
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering

LTN Academic Us

Summary

## LTN is **not** a special **type** of neural network

- it's a special way of training conventional NNs
- ▶ the NN still learns an optimal function
- ▶ optimality governed by knowledge axioms over data
- ▶ the optimal function learned by the NN is the one that best satisfies the knowledge axioms (given the data)

- ▶ LTN is **flexible** enough to handle **many** ML tasks
- LTN academic up-take has been limited, so far
- ► LTN reasoning capabilities are modest

### Introduct

- LTN Backgroun LTN Within NSI
- Logic Revi
- LTN Compone
- LTN Compone
- LTIN Real Logic
- LTN Knowledge
- LTN Loss Function
- LTN Training
- LTN Toy Example
- LTM Academic He

Summary

## LTN is **not** a special **type** of neural network

- it's a special way of training conventional NNs
- ▶ the NN still learns an optimal function
- optimality governed by knowledge axioms over data
- ▶ the optimal function learned by the NN is the one that best satisfies the knowledge axioms (given the data)

- ▶ LTN is **flexible** enough to handle **many** ML tasks
- LTN academic up-take has been limited, so far
- ► LTN reasoning capabilities are modest
- LTN has **scalability** limitations

### Introducti

- LTN Within NSI
- Logic Revie
- LTN Componer
- LTN Real Logic
- LTN Real Logic Axioms
- LTN Loss Function
- LTN Training
- LTN Toy Examples
  Github Repo Examples
  Binary Classification
  Clustering
- LTN Academic Us

Summary

## LTN is **not** a special **type** of neural network

- it's a special way of training conventional NNs
- the NN still learns an optimal function
- optimality governed by knowledge axioms over data
- ▶ the optimal function learned by the NN is the one that best satisfies the knowledge axioms (given the data)

- ► LTN is **flexible** enough to handle **many** ML tasks
- ▶ LTN academic **up-take** has been **limited**, so far
- ► LTN reasoning capabilities are modest
- LTN has **scalability** limitations
- ▶ LTN offers **explainability** via **querying** axiom satisfiability

# All Done

Introduction

LTN Backgroun

LTN Within NSI

\_\_\_\_\_

LTN Component

LITTE COMPONE

LTN Real Logic Axio

LTN Knowledge Base

LIN LOSS Func

LTN Training

LIN Toy Example

Binary Classification

LTN Academic Us

Summary

Thank You!