

I-Tensors

Implementations in `tnreason`

Logik für Erklärbare KI: Technische Einführung in das ENEXA Projekt

Maria Schnödt-Fuchs, Alex Goeßmann

Funded by the
European Union



15.+16. July, 2024

Demonstration package in python:

tnreason

tnreason = Tensor-Network Reasoning

Functionality:

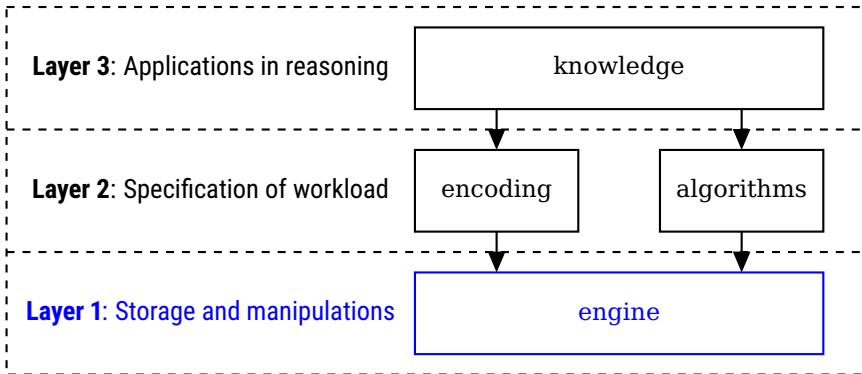
- ▶ Inference of Factored Systems based on logical and probabilistic concepts
- ▶ Neuro-Symbolic Reasoning Methods based on parametrized logics

tnreason is structured in four subpackages and three layers

- ▶ Layer 1: Storage and numerical manipulations, by subpackage engine , "Tensor Networks" -> building "tn" of tnreason
- ▶ Layer 2: Specification of workload, subpackage encoding specific for storage, subpackage algorithms specific for manipulations
- ▶ Layer 3: Applications in reasoning, by subpackage knowledge , "Reasoning" -> building "reason" of tnreason

Implementation in tnreason : Subpackage engine

The subpackage engine is dedicated to tensor networks and contractions.



Each Tensor core has attributes

- ▶ **values** (array-like): storing the value of the coordinates
- ▶ **colors** (list of str): specifying the name of the variables represented by its axes
- ▶ **name** (str): to distinguish from other cores

The implemented core types differ in the values argument. Cores are instantiated by

```
engine.getCore(coreType)(coreValues, coreColors, coreName)
```

Reflected in the notation

$$\mathcal{C}(\mathcal{T}^{\mathcal{G}}, \mathcal{V})$$

a contraction is defined by

- ▶ Tensor Network $\mathcal{T}^{\mathcal{G}}$, i.e. a dictionary of tensor cores
- ▶ Open Variables \mathcal{V}

Contraction calls are done by

`engine.contract(contractionMethod, coreDict, openColors)`

Where

- ▶ `contractionMethod`: str, chooses one of the contraction providers
- ▶ `coreDict`: Dictionary of TensorCores (of the above formats), representing the Tensor Network $\mathcal{T}^{\mathcal{G}}$
- ▶ `openColors`: List \mathcal{V} of str, each str identifying a color, that is a variable to be left open in the contraction

tnreason is maintained on github:

<https://github.com/EnexaProject/enexa-tensor-reasoning>

Installation using pip:

```
!pip install tnreason
```