On the design of a "Heavyweight" Artificial Neural Network Ontology

Fernando Lins

Advisor: Fred Freitas

Introduction

- Goals
  - Reason over Neural Networks
  - Enable the development of intelligent applications that understand and reason over Neural Networks
- Mainly MLP and Backpropagation
- Started with OWL and Protégé
- Problems with Math

Math in neural networks

- Activation function (Sigmoid)

$$f(x) = \frac{1}{1 + e^{-x}}$$

- Summation of entries

$$u = \sum_{j=1}^{N} x_j w_j$$

- Weight correction, etc

Mathematical problems

- OWL does not have sufficient expressivity to represent complex mathematical functions
- Some alternatives were considered to support the use of OWL, like:
  - SWRL (*Semantic Web Rule Language*) through built-in predicates
    - The use of built-in predicates is very limited to just some numerical operators, like add, divide, subtract and multiply
    - The built-in bridge extension supports the definition of user-defined built-ins, but it still has the problem that it is not integrated with the ontology and therefore you cannot reason over it

Mathematical problems

- MathML / OpenMath
  - Both are mathematical encodings for the web
  - OpenMath is aimed to encode the semantics of mathematics
  - MathML provide a presentation format for mathematical objects, and can have a pointer to some representation of the semantic of it
  - While meeting the needs of mathematical representation that we seek, this solution is still not desired because the representation is outside the ontology, thus preventing the reasoning
- EngMath
  - It is an ontology for mathematical modeling in engineering
  - Including conceptual foundations for quantities, dimensions and functions
  - The ontology itself is most concerned in defining the quantities and dimensions concepts, and uses the kif-numbers as a base for the functions

Mathematical problems

- After all these attempts, we've decided to use the expressivity of KIF (Knowledge Interchange Format), with the base of SUMO (Suggested Upper Merged Ontology)
- More specifically, we're using SUO-KIF
- And SIGMA as the Development Environment
- We plan to run it in Prolog afterwards… or in another suitable setting

SUMO

- Union of contributors (Engineering, Philosophy and Information Science) to develop an Upper Ontology (SUO – Standard Upper Ontology)
- Upper Ontology
  - Define generic and abstract concepts (High Level)
  - Not including domain-specific concepts (Engineering, Medicine…)
  - Provides the conceptual basis for domain and core ontologies
- Developed by the SUO Working Group, submitted and approved by IEEE
- Free and public

SUMO

- Uses material from the SUO's email list
- Syntactic and semantic merge of public available content into a single, comprehensive, and cohesive structure
  - Ontolingua server, John Sowa's upper level ontology, Ontologies developed by ITBM-CNR, Russell and Norvig's upper-level, etc
- Developed in SUO-KIF
- Proposed as initial document for SUO

SUO-KIF

- It was derived from KIF to support the definition of SUMO
- First-order against KIF`s high order
- Declarative semantics
- Logically comprehensive

SIGMA

- Sigma Knowledge Engineering Environment
- System for developing, viewing and debugging theories in first order logic
- Works with KIF
- Optimized for SUMO
- Features
  - Term and hierarchy browsing
  - Ability to load different files of logical theories
  - Full first order inference capability with structured proof results
  - Natural language paraphrase capability for logical axioms
  - Support for displaying mappings to the WordNet

Main classes

- Neuron
- Layer
- Learning algorithm
- Epoch
- Activation function
- Bias
- Phases
  - Training
  - Running
- Procedure

The Ontology: Neuron Structure

```
(instance Logistic UnaryFunction)
(instance Logistic TotalValuedRelation)
(domain Logistic 1 RealNumber)
(range Logistic MLPRealNumber)

(=>
   (equal (Logistic ?NUMBER1) ?NUMBER2)
   (equal ?NUMBER2 (DivisionFn 1 (AdditionFn 1 (ExponentiationFn NumberE (SubtractionFn 0 ?NUMBER1))))))
```

```
 ;; A Neuron has synapses that receives from itself and propagates to neuron of the next layer
(=> (instance ?NEURON Neuron)
    (exists (?SYNAPSE ?NEURONB ?LAYERA ?LAYERB)
            (and (hasSynapse ?NEURON ?SYNAPSE)
                 (receivesNeuron ?SYNAPSE ?NEURON)
                 (propagatesNeuron ?SYNAPSE ?NEURONB)
                 (neuronLayer ?NEURONB ?LAYERB)
                 (neuronLayer ?NEURON ?LAYERA)
                 (nextLayer ?LAYERA ?LAYERB)
                 (instance ?LAYERB Layer)
                 (instance ?LAYERA Layer)
                 (instance ?NEURONB Neuron)
                 (instance ?SYNAPSE Synapse)))))
```

The Ontology: MLP Structure

```
(instance hasLayer BinaryPredicate)
(domain hasLayer 1 MultilayerPerceptron)
(domain hasLayer 2 Layer)

(subrelation hasHiddenLayer hasLayer)
(domain hasHiddenLayer 1 MultilayerPerceptron)
(domain hasHiddenLayer 2 HiddenLayer)

(=>
   (and (instance ?MLP MultilayerPerceptron)
```

```
ayer Abstract)
nputLayer Layer)
iddenLayer Layer)
utputLayer Layer)
```

```
(instance nextLayer BinaryPredicate)
(instance nextLayer IrreflexiveRelation)
(domain nextLayer 1 Layer)
(domain nextLayer 2 Layer)

;The nextLayer of a HiddenLayer is another HiddenLayer or a OutputLayer
(=> (and (instance ?LAYERA HiddenLayer) (nextLayer ?LAYERA ?LAYERB))
    (or (instance ?LAYERB HiddenLayer) (instance ?LAYERB OutputLayer)))
```

The Ontology: Backpropagation

- Using the concepts of Procedure defined in SUMO
  - "A sequence-de
- Looking for a Co

```
(subclass Step Procedure)

(instance nextStep BinaryPredicate)
```

```
;Step has exactly one next Step
(=>
   (and (instance ?STEP Step)
        (not (instance ?STEP LastStep))
        (nextStep ?STEP ?NEXTSTEP))
   (equals (CardinalityFn ?NEXTSTEP) 1))

;;There is no step with a first step as next step
(=> (and (firstStep Procedure ?STEP) (instance ?STEP Step))
    (not (exists (?STEP2)
         (nextStep ?STEP2 ?STEP))))
```

Next steps

- Check ontology consistency against OntoClean
- Submit to ER 2010
- Translate it into Prolog (possibly using Ontolingua)
- Build applications

Thanks, questions??