neuronal symbolic integration

Josef Schulz

July 17, 2013

summary

neural symbolic circle

summary

neural symbolic circle

knowledge base

summary

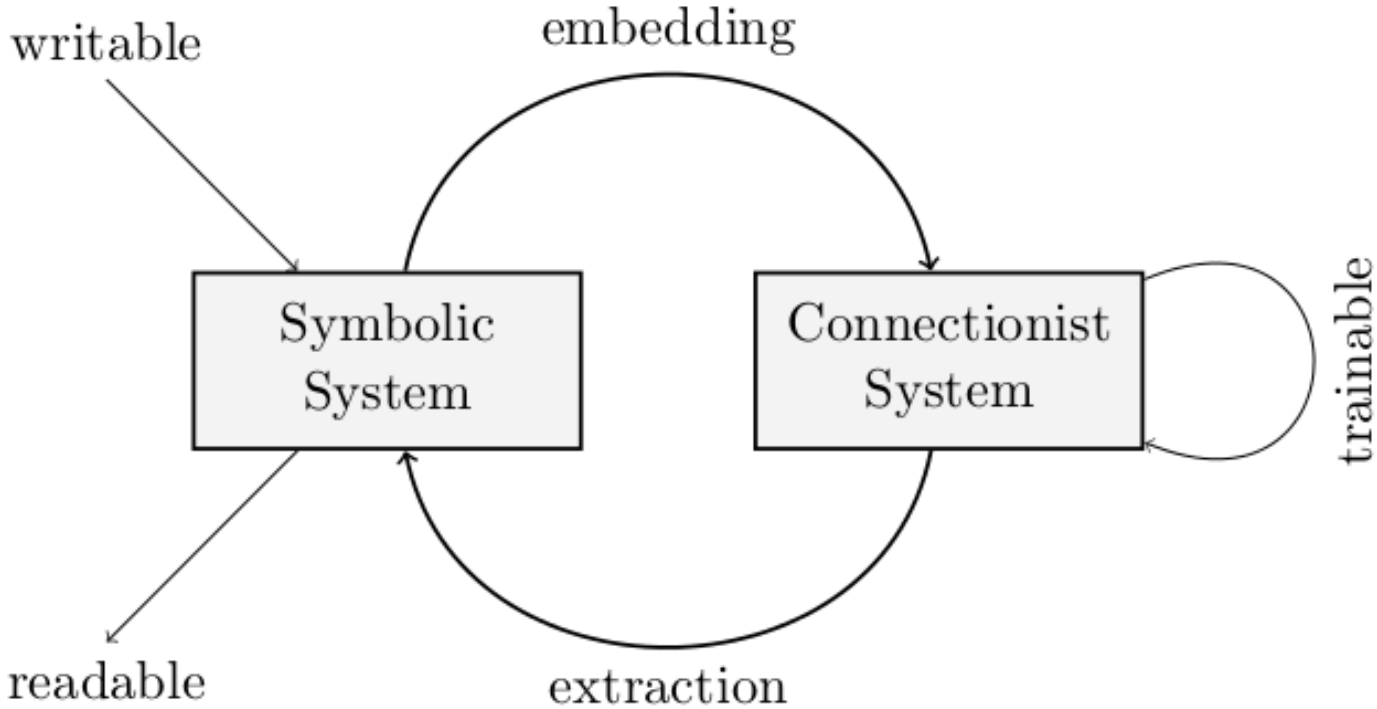
neural symbolic circle

knowledge base

embedding

summary

neural symbolic circle



knowledge base

The knowledge base is a set of propositional logic rules.
A subset of these rules will be named as a program.

A rule is defined as follows:
$A \leftarrow L_1 \wedge \ldots \wedge L_n$ and A is an atom
L_i with $i \in \{1, \ldots, n\}$ and $n \geq 0$ is a set of Literals

if n is equal to 0 the rule will be called as a fakt. $A \leftarrow \top$

Literals and Fakts

Negativ fakts are also possible: $A \leftarrow \bot$
It follows that the head of a rule is not realy an atom.

A is equal to $A \leftarrow \top$
$\sim A$ is equal to $A \leftarrow \bot$

Classical negation is also allowed, we call the programs extended.

$$v4 < -v3\ ,\ v8 < -v2\& \sim v1\ ,\ v15 < -v7\& \sim -v11$$

embedding

At first our network has just one hidden layer with 20 hidden units. I call this the basic units.

If we add a program, for each rule of the program another unit will be added to our hidden layer.

After this we update the number of basic units as follows:

$$numBasic = numBasic + numRules$$

train and add

Our network is constructed and we can learn it with backprobagation.

We are able to add more programs and every time after a program is added the network will be trained.

results

Trained with trainingset-10.txt
The file contains 13107 vector pairs.

numBasic	**numRules**	η	α	**time needed**	**steps needed**
23	0	0.4	0.3	9.720 s	11
17	5	0.4	0.3	5.740 s	5
17	5	0.4	0.3	8.180 s	8

The first is the empty program, the second a program with 5 correct rules and the last has also 5 rules but some of them are incorrect.

results

Trained with testset-65.txt
The file contains 89785 vector pairs.

numBasic	**numRules**	η	α	**time needed**	**steps needed**
23	0	0.4	0.3	26.950 s	5
17	5	0.4	0.3	11.950 s	2
17	5	0.4	0.3	22.980 s	4

The first is the empty program, the second a program with 5 correct rules and the last has also 5 rules but some of them are incorrect.

rule extraction

This is actually the third assignment.

I tried to implement the naive approach. The procedure is realy easy, I construct the truth table and for every variable the DNF could be extrakted.

At the end, the rules should be minimized with the Quine–McCluskey algorithm. But i have not done the last step..

rule extraction

After the network has been trained with the trainingset-10.txt i got 911358 rules in the following form:

$$v4 <- \sim 0\&v1\&v2\&v3\& \sim v4\&v5\&v6\& \sim v7\&\ v8\&v9\& \sim v10\&v11\& \sim v12\&v13\&v14\&v15\&v16$$

questions

the hole program is written in C++