

Differentiable Neural Computer Learning Note

Zhengzhong Liang

2018-04-30

1 Learning Note 20180505

1.1 Agent Task Analysis

1.1.1 What is the agent trying to learn?

This agent tries to learn a pattern with timestep 9 and dimension 6 (according to the default print result). This pattern is generated by input sequence, which is tensor with shape $9 \times 16 \times 6$. So I

1.1.2 Is there any law used to generate this pattern?

1.2 Code Features

2 Code Structure

2.1 About Computation Graph

The basic graph is defined in “DNC.py”, some peripheral parts of graph is defined in “run_model” function and “train.py”. The main parts of computation graph of DNC is organized as following:

2.2 Activities in Each Loop

In each loop of training in iteration, the agent does the following things:

Some input is presented

Input is given to LSTM

Something is written to memory

Something is output from memory

The output is decoded by LSTM and printed

Table 1: Organization of DNC

name	functionality	script	function	class
controller	an LSTM network which reads data	DNC.py	x	DNC
access	the memory of DNC	access.py,		

Table 2: The Structure and Size of State Tuple

Level 1	Level 2	Level 3	Level 4
DNCState	access_output ($16 \times 4 \times 16$)		
	access_state	memory ($16 \times 16 \times 16$)	
		read_weights ($16 \times 4 \times 16$)	
		write_weights ($16 \times 1 \times 16$)	
		linkage	link ($16 \times 1 \times 16 \times 16$)
			precedence_weights ($16 \times 1 \times 16$)
		usage (16×16)	
	controller_state	hidden (16×64)	
		cell (16×64)	

Table 3: Important Parameters (from Code)

item	value	meaning
hidden size	64	The number of neurons of LSTM (controller)
memory size	16	the number of memory slots
word size	16	the length of each memory slot
num write heads	1	the number of write heads
num read heads	4	the number of read heads

3 Topics to Study

3.1 What type of task does this agent try to finish

3.2 What is the function of memory in this task

3.3 What tyep of tasks can we train the agent to do

3.4 What parts and functions can we add to this agent

4 Code Features

4.1 The Structure of Return Value of DNC Core

5 Question

5.1 The memory and controller are defined, but are not connected

They are connected in the build function, which is called by dynamic rnn function.

Table 4: Tensor Size and Explanation

tensor name	tensor size	explanation
access output	$16 \times 4 \times 16$	batch size, # of read heads, # of memory slots
memory	$16 \times 16 \times 16$	batch size, # of memory slots, slot length
read weights	$16 \times 4 \times 16$	batch size, # of read heads, # of memory slots
write heads	$16 \times 4 \times 16$	batch size, # of writes heads, # of memory slots
link		
precedence weights		
usage	16×16	batch size, # of memory slots
hidden	16×64	batch size, # of LSTM neurons
cell	16×64	batch size, # of LSTM neurons

5.2 The build function is not used

They are used in dynamic rnn function.

5.3 The functionality of dynamic rnn

It connects the different parts of the computation graph of DNC, so that a complete graph will be established after calling this function.

5.4 In every loop a new DNC is created?

No, the DNC is only built once.

5.5 How the computation graph is built in tensorflow?

Anything before a session starts is considered as computation graph.

5.6 What does build do in the code

Connects different parts of computations graph. Build function should be originally a method of `snt.AbstractModule`. The abstract module is define [here](#).

As one can see, the build function is a method in this class.

5.7 Draw UML for the code

See the UML diagram in the folder.

5.8 Why there is a init function and a build function in each class

init function creates the components of the computation graph, while the build function connects these components. The init function is called when the object of DNC is initiated. The build function is called in the `tf.nn.dynamic_rnn` function.

5.9 The sequence of calling of build function

DNC, Access, Freeness, ConsineWeights, TemporalLinkage, ConsineWeights.

5.10 Draw the inheritance relation ship (and attributes) starting from AbstractModule

5.11 Figure out how abstract class works in python (e.g. why no init function)

Question resolved.

5.12 Another implementation of DNC

[Is available here](#)

5.13 Is the input presented to DNC at each time step?

5.14 What is the timestep of DNC

Since all state tensors have 16 in its dimension, the timestep seems to be 16.

5.15 What is the size of each memory block in DNC memory?

It should be either 16 (because the last dimension in access_output is 16) or 64 (because the LSTM size is 64).

6 Questions about DNC

6.1 Equivalence between Neural network, Turing machine and brain?

It is believed that RNN is Turing Complete (Turing complete seems mean that something is equivalent to Turing machine.). And it is believed by some people that brain is not more powerful than Turing machine. So that we can conclude that $\text{brain} \neq \text{RNN} = \text{Turing Machine}$.

However, it is questionable then that why we need external memory attached to RNN if RNN alone is Turing complete.