

Differentiable Neural Computer Learning Note

Zhengzhong Liang

2018-04-30

1 Code Structure

1.1 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:

1.2 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,		

2 Topics to Study

- 2.1 What type of task does this agent try to finish**
- 2.2 What is the function of memory in this task**
- 2.3 What together tasks can we train the agent to do**
- 2.4 What parts and functions can we add to this agent**

3 Question

- 3.1 The memory and controller are defined, but are not connected**

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

- 3.2 The build function is not used**

They are used in dynamic rnn function.

- 3.3 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.

- 3.4 In every loop a new DNC is created?**

No, the DNC is only built once.

- 3.5 How the computation graph is built in tensorflow?**

Anything before a session starts is considered as computation graph.

- 3.6 What does build do in the code**

- 3.7 Draw UML for the code**

- 3.8 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.