# 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, | | |

| Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|
| DNCState | access_output | | |
| | access_state | memory | |
| | | read_weights | |
| | | write_weights | |
| | | linkage | link |
| | | | precedence_weights |
| | | usage | |
| | controller_state | hidden | |
| | | cell | |

# 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 tyep of tasks can we train the agent to do

## 2.4 What parts and functions can we add to this agent

# 3 Code Features

## 3.1 The Structure of Return Value of DNC Core

# 4 Question

## 4.1 The memory and controller are defined, but are not connected

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

## 4.2 The build function is not used

They are used in dynamic rnn function.

## 4.3 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.

## 4.4 In every loop a new DNC is created?

No, the DNC is only built once.

### 4.5 How the computation graph is built in tensorflow?

Anything before a session starts is considered as computation graph.

### 4.6 What does build do in the code

Connects different parts of computations graph. Build function should be orginally a method of snt.AbstractModule. The abstract module is define here.

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

### 4.7 Draw UML for the code

See the UML diagram in the folder.

### 4.8 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.

### 4.9 The sequence of calling of build function

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

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

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

Question resolved.

### 4.12 Another implementation of DNC

Is available here

### 4.13 Is the input presented to DNC at each time step?

### 4.14 What is the timestep of DNC

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

### 4.15 What is the size of each memory block in DNC memory?

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

## 5 Questions about DNC

### 5.1 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 brain≠RNN=Turing Machine.

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