Neural Reasoning

INTELLIGENT SYSTEMS FOR PATTERN RECOGNITION (ISPR)
Antonio Carta – antonio.carta@unipi.it
DAVIDE BACCIU – DIPARTIMENTO DI INFORMATICA - UNIVERSITA’ DI PISA
DAVIDE.BACCIU@DI.UNIPI.IT
Neural Reasoning

- Recurrent (sequential) models as general programs
- Reasoning abilities, typical of classic algorithms or classic AI
- Why don’t we use a «classic» algorithm? Because we may not have a proper input (e.g. pathfinding graph) but only sensor information.
  - Example: robot navigation
  - The model needs to learn to encode the structure from the raw data and then solve the problem
Example: Pathfinding
RNN and Memory – Issue 2

A motivating example:

<table>
<thead>
<tr>
<th>Task 3: Three Supporting Facts</th>
<th>Task 15: Basic Deduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>John picked up the apple.</td>
<td>Sheep are afraid of wolves.</td>
</tr>
<tr>
<td>John went to the office.</td>
<td>Cats are afraid of dogs.</td>
</tr>
<tr>
<td>John went to the kitchen.</td>
<td>Mice are afraid of cats.</td>
</tr>
<tr>
<td>John dropped the apple.</td>
<td>Gertrude is a sheep.</td>
</tr>
<tr>
<td>Where was the apple before the kitchen? A: office</td>
<td>What is Gertrude afraid of? A: wolves</td>
</tr>
</tbody>
</table>

○ In order to solve the task need to memorize
  ● Facts
  ● Question
  ● Answers

○ A bit too much for the dynamical RNN memory

○ Try to address it through an external memory
Memory Networks - General Idea

Neural network

writing

Memory module

reading
Memory Network Components

(I) **Input feature map**: Encodes the input in a feature vector

(G) **Generalization**: decide what input (or function of it) to write to memory

(O) **Output feature map**: reads the relevant memory slots

(R) **Response**: returns the prediction given the retrieved memories

Weston et al, Memory Networks, ICLR 2015
End-to-End Memory Networks

Combine output memories through soft attention

Facts $x_i$

Search for memories matching the query

Generate prediction

Query driven soft-attention

Sukhbaatar et al, End-to-end Memory Networks, NIPS 2015
Memory Network Extensions

Often with tied weights

Use more complex output components, e.g. RNN to generate response sequences

Stack multiple memory network layers

Several iterations of reasoning to produce a better answer
Memory Nets for Visual QA with Attention

Yang et al, Stacked Attention Networks for Image Question Answering, CVPR 2016
Memory Nets for Visual QA with Attention

(a) What are pulling a man on a wagon down on dirt road?  
   Answer: horses  Prediction: horses

(b) What is the color of the box?  
   Answer: red  Prediction: red

(c) What next to the large umbrella attached to a table?  
   Answer: trees  Prediction: tree

(d) How many people are going up the mountain with walking sticks?  
   Answer: four  Prediction: four

(e) What is sitting on the handle bar of a bicycle?  
   Answer: bird  Prediction: bird

(f) What is the color of the horns?  
   Answer: red  Prediction: red
Neural Turing Machines

- Memory networks that can read and write memories at both training and test
- End-to-end differentiable
Neural Controller

Typically an RNN emitting vectors to control read and write from the memory

The key to differentiability is to always read and write the whole memory

Use soft-attention to determine how much to read/write from each point
Memory Read

Interest in the i-th memory

Attention distribution vector $\mathbf{a}$ from the RNN

Memories $\mathbf{M}_i$

Retrieved memory is a weighted mean of all memories

$$\mathbf{r} = \sum_i a_i \mathbf{M}_i$$
Memory Write

Value to write \( w \)

Attention distribution vector \( \alpha \) describing how much we change each memory

\[ M_i = a_i w + (1 - a_i) M_i \]

Write operation is actually performed by composing erasing and adding operations
NTM Attention Focusing

1. Generate content-based memory indexing

- Memory
- Query vector

```
[Memory]
\rightarrow dot product
\rightarrow softmax

[Output]
```

A
NTM Attention Focusing

2. Interpolate with attention from previous time

A

interpolate

B

Previous attention vector

Interpolation amount
NTM Attention Focusing

3. Generate location-based indexing

B

Shift distribution filter

Determines how we move between the locations in memory

Sharpen the distribution for final memory access
Practical Use?

○ Not yet..

○ Not straightforward to train

○ Advantages over GRNN when it comes to learn to program
  ● Copy task
  ● Repeat copy
  ● Associative recall
  ● Sorting

○ Foundations for neural reasoning
  ● Pondering networks
Software

- Complete sequence-to-sequence tutorial (including attention) on **Tensorflow**
  - A shorter version in **Keras**
- **Github project** collecting several memory augmented networks
- **Pytorch** implementation of stacked attention for visual QA (originally Theano-based)
- Many implementations of the NTM (Keras, Pytorch, Lasagne,...): none seemingly official, but **this recent one** is supposedly stable (enough for TF to list it as official)
Take Home Messages

- Attention.. Attention.. and, again, attention
  - Soft attention is nice because makes everything fully differentiable
  - Hard attention is stochastic hence cannot Backprop
  - Empirical evidences of them being sensitive to different things

- Encoder-Decoder scheme
  - A general architecture to compose heterogeneous models and data
  - Decoding allows sampling complex predictions from an encoding conditioned distribution

- Memory and RNN
  - Efficient use of dynamic memory
  - External memory for search and recall tasks
  - Read/write memory for neural reasoning