Memory Networks

Zhong Wenjie
zvengin@nii.ac.jp

April 11, 2018

1 General Introduction

1.1 Problem Definition

Most machine learning models lack an easy way to read and write a long-term memory component, and to combine this with inference seamlessly. This paper propose memory framework which reasons with an inference component combined with a long-term memory component. They learn how to operate these jointly. Although emergency of RNN provide possibility to some tasks that require memory, RNN has a difficulty in remembering long-term information since its small memory and it is not compartmentalized enough to remember past facts exactly.

1.2 General Framework

General frame work of memory network consists of five components including I component, which maps inputs to internalfeatures, G component, which is responsible for updating memory, O component,which outputs related information, and R component, which converts outputs from O component to the format we desired.
G component: $m_{H(x)} = x$
here $H(.)$ is a function maps inputs to index in memory. G will update memory which $H(.)$ indicates. complex G component may go back and update previously stored memories. If memory is full, $H(.)$ function can decide which memory should be replaced.

2 An implementation of Memory Network for Text

2.1 I component

Here I component is a function which maps input text to a D dimention bag of words, $I = \phi(x)$.

2.2 G component

Here G component is simply a function which write input features to the next available memory without updating old memory.

2.3 O compoent

O component is core of inference in this task. This component will select related information from memory to generate output text and selection mechanism is based on score function $S_O(x, y) = \phi_x(x)U^TU\phi_y(y)$. Here $U \in R^{n*D}$ is embedding matrix which encodes input feature into n dimention. selected information $m_1 = argmax_{m_i} S_O(x, m_i), i = 1, ..., M$.

2.4 R component

R component generate text according selected information. Here this paper only outputs one word based on score function $S_R(x, y) = \phi_x(x)U^TU\phi_y(y)$. output word $w = argmax_{w_i} S_R([x, m_1, m_2], w_i)$

2.5 Training

This model is trained in a supervised setting in which we are given correct target and supporting evidence. The objective of this task is minimizing margin rank loss, $Argmin_{U_O,U_R} \sum_{m_{o1} \neq m_f} max(0, \gamma - S_O(x, m_{o1}) + S_O(x, m_f)) + \sum_{m_{o2} \neq m_f} max(0, \gamma - S_O([x, m_{o1}], m_{o2}) + S_O([x, m_{o1}], m_f)) + \sum_{r \neq r'} max(0, \gamma - S_R([x, m_{o1}, m_{o2}], r) + S_O([x, m_{o1}, m_{o2}], r'))$ Here m_{o1}, m_{o2}, r are supporting evidence and correct target in training set and m_f, r' are other choices other than correct evidence and target.