CS 536 Machine Learning II: Final Project Report
Team No. 20

Harshith T (ht354), Desmond L (dl1053), Jhanavi B(jb2040), Tanvi S (tas302), Tarini D (td508), Pruthviraj P (pp865)

Abstract

The ability to reason and answer questions is a characteristic of neural network architectures that use memory and attention mechanisms. The dynamic memory network (DMN) is one such architecture that has shown high accuracy on language tasks. However, it is unclear whether the DMN performs well on question-answering tasks without marked supporting facts during training, or whether it can be applied to other modalities, such as images.

To address these questions, we have implemented a DMN model and proposed improvements to its memory and input modules. A novel image input module has been introduced, enabling the DMN to answer visual questions. The new DMN+ model surpasses state-of-the-art on both the bAbI-10k text question-answering dataset and the Visual Question Answering dataset, even without supporting fact supervision.

1. Introduction

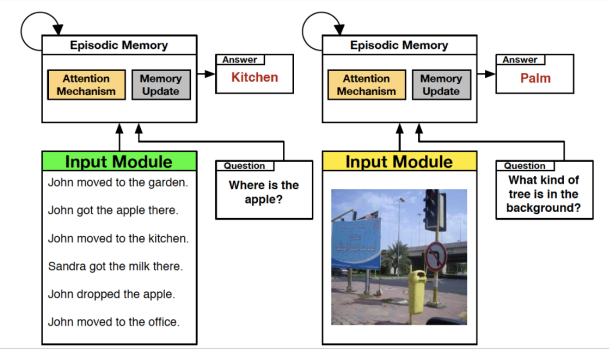

Figure 1. Text/Visual input and output

This project implements improvements to the input and memory modules of DMN, which are essential for enhancing question answering. A two-level encoder with a sentence reader and input fusion layer is used for the input module to allow information flow between sentences. The memory module is modified with attention gates to select important facts so that DMN+ no longer requires labeled supporting facts during training. A new image input module is introduced, and the modified memory module improves both textual and visual question answering.

2. Dynamic Memory Networks

The DMN is a general architecture for question answering (QA). It is composed of modules that allow different aspects such as input representations or memory components to be analyzed and improved independently.GRUs are the basic building blocks of our DMN. For each time step i with input x_i and previous hidden state h_{i-1} we compute new hidden state h_i, σ is the sigmoid activation function.

$$z_i = \sigma(W^{(z)}x_i + U^{(z)}h_{i-1} + b^{(z)})$$
$$r_i = \sigma(W^{(r)}x_i + U^{(r)}h_{i-1} + b^{(r)})$$
$$\hat{h}_i = tanh(Wx_i + r_i \circ Uh_{i-1} + b^{(h)})$$
$$h_i = u_i \circ \hat{h}_i + (1 - u_i) \circ h_{i-1}$$

2.1. Input module

This module processes the input data about which a question is being asked into a set of vectors termed facts, represented as$F = [f_1, f_2, \ldots f_N]$, where N is the total number of facts. These vectors are ordered, resulting in additional information that can be used by later components.

2.2. Question module

The vector representation ”q” of the question is computed by this module, utilizing the final hidden state of a GRU (Gated Recurrent Unit) that operates over the words in the question.

2.3. Episodic Memory module

The retrieval of necessary information from input facts to answer a given question, known as episode memory, involves

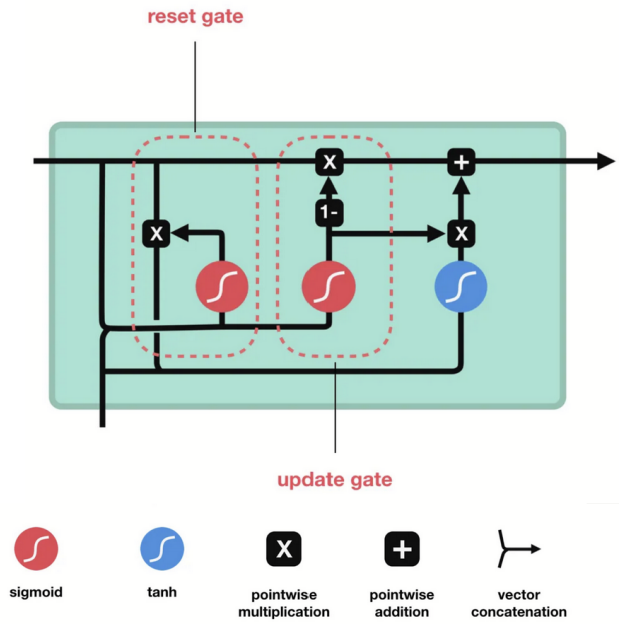

Figure 2. GRU

multiple passes over the input to enhance understanding of both the question and input, particularly for questions requiring transitive reasoning. The initial memory vector is set to the question vector, and the episode memory on the t^{th} pass over the inputs is referred to as m^t, where m^t The episodic memory module comprises two distinct components: the attention mechanism and the memory update mechanism. The attention mechanism generates a contextual vector c^t, which is a summary of relevant input for pass t, with relevance inferred by the question q and previous episode memory m^{t1}. The memory update mechanism generates the episode memory mt based on the contextual vector c^t and previous episode memory m^{t1}. The final pass T should result in the episodic memory m^T containing all the information necessary to answer the question q.

2.4. Answer module

The answer module uses q and m^T to generate the predicted answer. For simple answers, a linear layer with softmax activation may be used. For sequence outputs, an RNN can decode $[q; m^T]$ to tokens.

3. DMN+ - Modified Input Module : Embeddings

In the DMN+ model, the two different components which replace single GRU are sentence reader and input fusion layer. The sentence reader encodes words into a sentence

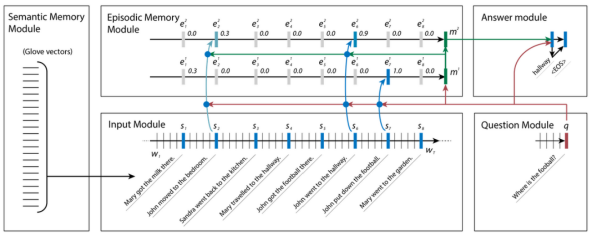

Figure 3. DMN architecture

embedding. The input fusion layer enables interactions between sentences. We use a bi-directional GRU in the input fusion layer because it allows information from both past and future sentences to be used. The input module uses positional encoding for the sentence reader and a bi-directional GRU for the input fusion layer. The input fusion layer applies a bi-directional GRU, allowing contextual information from both future and past facts to impact the current input fact.

3.1. Input module for Text QA

In order to represent a sentence in the positional encoding scheme, we sum the element-wise multiplication of each word embedding with a corresponding vector l_j

$$f_i = \sum_{M}^{j=1} l_j \circ w_j^i$$

$$l_{jd} = (\frac{1-j}{m}) - (\frac{d}{D})(\frac{1-2j}{M})$$

where l_j is a column vector, d is embedding index , D is the dimension of embedding.

These facts are passed to the bidirectional GRUs

$$\overrightarrow{f_i} = GRU_{fwd}(f_i, \overrightarrow{f_{i-1}})$$
$$\overleftarrow{f_i} = GRU_{bwd}(f_i, \overleftarrow{f_{i+1}})$$
$$\overleftrightarrow{f_i} = \overleftarrow{f_i} + \overrightarrow{f_i}$$

3.2. Input module for Visual QA

We use convolutional neural network to extract image features. Prior to feeding the image into the network, we resize it to 448 × 448 pixels. We extract features from the last pooling layer which produces 512-dimensional output for each

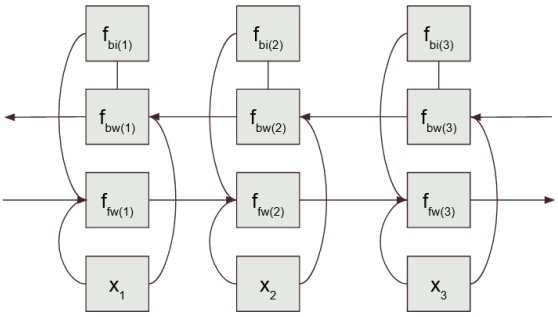

Figure 4. Bi-directional GRU

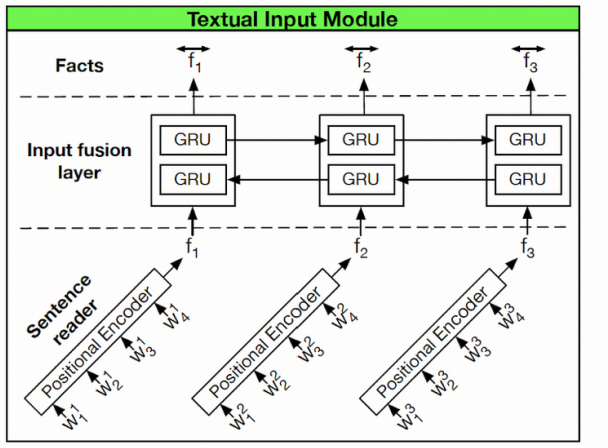

Figure 5. Textual input module

of the 14 × 14 grids. In total, this results in 196 regional feature vectors of 512 dimensions.

3.3. Episodic Memory Module

The episodic memory module retrieves information from the input facts by associating a single scalar value, the attention gate g_i^t , with each fact f_i during pass t.

$$z_i^t = [\overleftrightarrow{f_i} \circ q; \overleftrightarrow{f_i} \circ m^{t-1}; |\overleftrightarrow{f_i} - q|; |\overleftrightarrow{f_i} - m^{t-1}|]$$

$$Z_i^t = W^{(2)} tanh(W^{(1)} z_i^t + b^{(1)}) + b^{(2)}$$

$$g_i^t = \frac{exp(Z_i^t)}{\sum_{k=1}^{M_i} exp(Z_k^t)}$$

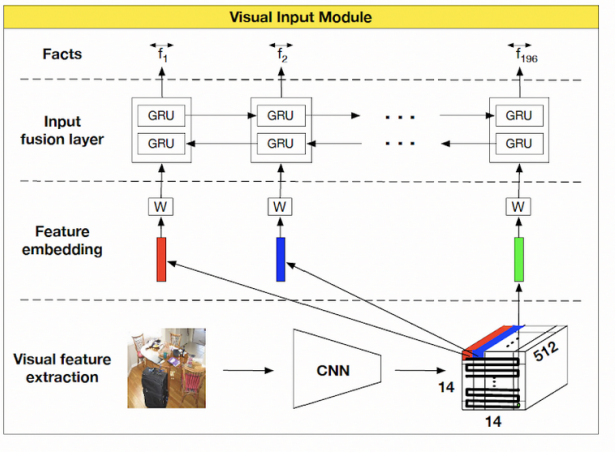

Figure 6. Visual input module

4. Attention Mechanism

After obtaining the attention gate g_i^t, we utilize an attention mechanism to obtain a contextual vector c^t, which is centered on the current focus

4.1. Soft Attention

$$c^t = \sum_{i=1}^{N} g_i^t \overleftrightarrow{f_i}$$

We calculate the contextual vector ct by weighted summation of vectors from the input facts F and their corresponding attention gates.It is easy to compute and can approximate hard attention with differentiability.However, this method loses positional and ordering information due to summation, which may require multiple passes to retrieve efficiently.

4.2. Attention Based GRU

$$h_i = g_i^t \circ \tilde{h_i} + (1 - g_i^t) \circ h_{i-1}$$

For complex queries, we need the attention mechanism to consider both position and ordering of input facts. RNNs are not suitable for this task as they cannot utilize the attention gate information .So to tackle this we modified the GRU architecture by embedding information from the attention mechanism.By replacing the update gate in GRU with the output of the attention gate, the GRU can use the attention gate for updating its internal state. The attention gate is a scalar generated using softmax activation, allowing for easy visualization of activation over input.The contextual vector c^t used for updating the episodic memory state is obtained from the final hidden state of the attention-based GRU.

5. Episodic Memory Updates

$$m^t = ReLU(w^t[m^{t-1}; c^t; q] + b)$$

Episodic memory m^{t-1} is updated with the contextual vector c^t after each pass through the attention mechanism.A GRU with the initial hidden state set to the question vector q is used in the DMN to update the episodic memory m^t for pass t.Using different weights for each pass through the episodic memory can be advantageous.A Rectified Linear Unit (ReLU) layer is used for the memory update. This involves concatenating the previous episode memory (m^{t-1}), the contextual vector (c^t), and the question vector (q), passing it through a ReLU activation, and then applying a weight matrix ($W^t t$) and bias vector (b). This modification improves accuracy .The final output of the memory network is then passed to the answer module.

6. Dataset

6.1. BABI-10K

Its a dataset which consists of 20 tasks each of which contain 10,000 training samples and 1000 test samples.These tasks consist of facts, a question, the answer, and supporting facts that provide a path to the answer.The tasks are designed to evaluate a system's ability to understand and answer questions through various methods such as chaining facts, induction, and deduction

6.2. Visual Question Answering (VQA) dataset

The Visual Question Answering (VQA) dataset is constructed using the Microsoft COCO dataset.It has 123,287 training/validation images and 81,434 test images.Each image has several related questions with each question answered by multiple people. It has in total 248,349 training questions, 121,512 validation questions, and 244,302 for testing

7. Results

Below are the results of our model on BABI-10 k dataset. Our model performed really well for tasks such as supporting facts, yes/no question, counting which require chaining of facts. But it performed poor for complex tasks such as induction and positional reasoning.Also comparing error rates of our model to the error rates of previous DMN model showcased that our model performed better than the old models in most of the tasks

Task	Task	Training accuracy	Test accuracy
1	1 supporting fact	76.1	74.9
2	2 supporting facts	99.7	99.4
3	3 supporting facts	95.3	94.1
5	3 argument relations	99.5	99.4
6	yes / no questions	99.8	99.7
7	counting	98.7	98.6
14	Time Reasoning	99.2	98.9
16	Basic Induction	52.3	51.2
17	Positional Reasoning	12.7	12.4
18	Size Reasoning	93.2	92.5

Task number	ODMN	DMN2	DMN3	DMN+
1	NA	NA	NA	25.1
2	36.0	0.1	0.7	0.6
3	42.2	19.0	9.2	5.9
5	0.1	0.5	0.8	0.6
6	35.7	0.0	0.6	0.3
7	8.0	2.5	1.6	1.4
14	3.6	0.7	0.0	1.1

8. Conclusions

- The model replaces the input module with an input fusion layer, which improves interaction between distant facts in both the textual and visual datasets.
- It replaces the soft attention mechanism with an attention-based GRU, which helps answer questions where complex positional or ordering information may be required. This change impacts the textual dataset the most.
- It increases the performance by using unique set of weights for each pass and a linear layer with a ReLU activation to compute the memory update
- The improved performance of the DMN+ model shows that the DMN framework can be generalized across input domains.
- The proposed model changes allow for logical reasoning over ordered inputs and interactions between input facts, which are important for complex reasoning tasks.

9. References

(1) C. Xiong, S. Merity, R. Socher, "Dynamic Memory Networks for Visual and Textual Question Answering", arXiv:1603.01417v1, 2016.

(2) A. Agrawal, J. Lu, S. Antol, M. Mitchell, C.L. Zitnick, D. Batra, D. Parikh, "VQA: Visual Question Answering", arXiv:1505.00468v7, 2016

(3) X. Chen, C.L. Zitnick, "Learning a Recurrent Visual Representation for Image Caption

Generation",arXiv:1411.5654v1, 2014

(4) K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", EMNLP, 2014.

(5) A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, R. Socher, "Ask Me Anything: Dynamic Memory Networks for Natural Language Processing", arXiv:1506.07285, 2016

(6) J. Li, M.T. Luong, D. Jurafsky, "A Hierarchical Neural Autoencoder for Paragraphs and Documents", arXiv:1506.01057, 2015

(7) T.Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C.L. Zitnick, P. Dollár, "Microsoft COCO: Common Objects in Context", arXiv:1405.0312, 2015

(8) M.T. Luong, H. Pham, C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation", arXiv:1508.04025, 2015

(9) J. Weston, S. Chopra, A. Bordes, "Memory Networks", arXiv:1410.3916, 2015

(10) M. Stollenga, J. Masci, F. Gomez, J. Schmidhuber, "Deep Networks with Internal Selective Attention through Feedback Connections", ICML 2014