

A Survey on Neural Networks Using Memory Component

Jihwan Lee[†] · Jinuk Park^{**} · Jaehyung Kim^{***} · Jaein Kim^{****} ·
Hongchan Roh^{*****} · Sanghyun Park^{*****}

ABSTRACT

Recently, recurrent neural networks have been attracting attention in solving prediction problem of sequential data through structure considering time dependency. However, as the time step of sequential data increases, the problem of the gradient vanishing is occurred. Long short-term memory models have been proposed to solve this problem, but there is a limit to storing a lot of data and preserving it for a long time. Therefore, research on memory-augmented neural network (MANN), which is a learning model using recurrent neural networks and memory elements, has been actively conducted. In this paper, we describe the structure and characteristics of MANN models that emerged as a hot topic in deep learning field and present the latest techniques and future research that utilize MANN.

Keywords : Recurrent Neural Networks, Memory-Augmented Neural Networks, Memory Component, Memory Networks, Neural Turing Machines, Stack-Augmented Neural Networks

메모리 요소를 활용한 신경망 연구 동향

이 지 환[†] · 박 진 욱^{**} · 김 재 형^{***} · 김 재 인^{****} · 노 흥 찬^{*****} · 박 상 현^{*****}

요 약

최근 순환 신경 망(Recurrent Neural Networks)은 시간에 대한 의존성을 고려한 구조를 통해 순차 데이터(Sequential data)의 예측 문제 해결에서 각광받고 있다. 하지만 순차 데이터의 시간 스텝이 늘어남에 따라 발생하는 그라디언트 소실(Gradients vanishing)이 문제로 대두되었다. 이를 해결하기 위해 장단기 기억 모델(Long Short-Term Memory)이 제안되었지만, 많은 데이터를 저장하고 장기간 보존하는 데에 한계가 있다. 따라서 순환 신경망과 메모리 요소(Memory component)를 활용한 학습 모델인 메모리-증대 신경망(Memory-Augmented Neural Networks)에 대한 연구가 최근 활발히 진행되고 있다. 본 논문에서는 딥 러닝(Deep Learning) 분야의 화두로 떠오른 메모리-증대 신경망 주요 모델들의 구조와 특징을 열거하고, 이를 활용한 최신 기법들과 향후 연구 방향을 제시한다.

키워드 : 순환 신경망, 메모리-증대 신경망, 메모리 요소, 메모리 네트워크, 뉴럴 튜링 머신, 스택-증대 신경망

1. 서 론

하루에도 엄청난 양의 데이터가 생겨나는 빅데이터 시대에서, 하드웨어의 발전에 의해 대용량의 데이터를 처리하고 학습시키는게 가능해져 인공지능 분야의 발전이 급속도로 이루어지고 있다. 이에 따라, 입력 값에 의해 생성된 출력 값이

이후 학습에 영향을 주는 순환 신경망(Recurrent Neural Networks)[1]이 주목받고 있다. 순환 신경망은 시간에 대한 의존성을 요구하고 순차 데이터를 사용하는 학습 모형으로 널리 활용되고 있다.

순환 신경망은 신경망의 깊이가 깊어짐에 따라 오차를 줄이는 학습이 어려운 그라디언트 소실의 치명적인 단점이 있다. 즉, 입력 데이터의 양이 늘어남에 따라 오래된 입력 데이터에 대한 기억 유지가 어렵다. 이를 해결하기 위해 순환 신경망 구조를 기반으로 셀 스테이트(Cell state)라는 기억 공간을 둔 장단기 기억 모델[2]이 제안되었다.

장단기 기억 모델은 셀 스테이트와 데이터의 기억 삽입/유지/삭제를 위한 게이트(gate)를 통해 순환 신경망의 그라디언트 소실 문제점을 해결할 수 있었다. 하지만 장기 기억 의존성(Long-term dependency)을 요구하는 심화 문제에서 입력 데이터에 대한 기억의 보존성과 지속성에서 한계를 보였다.

※ 이 논문은 2015년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2015R1A2A1A05001845).

† 비 회 원 : 연세대학교 컴퓨터과학과 석사과정

** 준 회 원 : 연세대학교 컴퓨터과학과 석사과정

*** 비 회 원 : 연세대학교 컴퓨터과학과 박사과정

**** 비 회 원 : 연세대학교 컴퓨터과학과 학사과정

***** 비 회 원 : SK Telecom ICK 종합원 매니저

***** 중신회원 : 연세대학교 컴퓨터과학과 교수

Manuscript Received : February 8, 2018

First Revision : April 13, 2018

Accepted : May 3, 2018

* Corresponding Author : Sanghyun Park(sanghyun@yonsei.ac.kr)

따라서 순환 신경망 기반의 모델이 입력된 데이터의 기억에 대한 보존성과 지속성을 갖기 위해 메모리 요소의 필요성이 대두되었다. 메모리 요소는 신경망 밖에 위치하여 데이터를 저장하고, 메모리 요소에 저장된 데이터를 바탕으로 신경망과 상호작용하여 문제를 해결할 수 있다는 이점을 갖는다.

최근 순환 신경망이 메모리 요소와 상호 작용할 수 있도록 고안된 모델인 메모리-증대 신경망(Memory-Augmented Neural Networks)에 대한 연구가 각광받고 있다. 메모리-증대 신경망은 메모리 네트워크 (Memory Networks), 뉴럴 튜링 머신(Neural Turing Machines), 그리고 스택-증대 신경망(Stack-Augmented Neural Networks)으로, 크게 세 가지로 분류할 수 있다[3-6].

메모리 네트워크[3]는 자연어처리(Natural Language Processing)에서 복잡한 문제로 거론되는 질의-응답 문제(Question-Answering problem)를 해결하기 위한 목적으로 제안된 모델이다. 메모리 요소에 저장된 데이터를 바탕으로 질의-응답 문제에서 답을 도출하기 위한 추론 능력이 뛰어난 구조를 갖는 것이 큰 특징이다.

뉴럴 튜링 머신[4]은 튜링 머신(Turing Machine)의 구조를 착안하여 고안된 컴퓨팅 모델이다. 뉴럴 튜링 머신은 추가한 메모리 요소 속의 특정 위치를 선택하여 읽기 또는 쓰기를 수행하도록 학습하는 주소지정 메커니즘(Address mechanism)를 사용한다. 따라서 뉴럴 튜링 머신의 메모리 접근 방식은 메모리 네트워크에서 사용되는 방식과 다르다. 이러한 메모리 접근 방식을 통해 복잡한 알고리즘적 문제 해결에 대한 학습이 가능함을 보였다.

스택-증대 신경망[5, 6]은 신경망 모델과 스택(Stack), 큐(Queue), 덱(Deque), 혹은 리스트(List)과 같은 자료구조에 데이터를 저장하고 신경망과 상호작용하는 모델이다. 스택-증대 신경망은 자료구조 형태의 구조화된 메모리 요소를 사용하는 메모리-증대 신경망의 또 다른 연구라 할 수 있다. 스택-증대 신경망은 순차 데이터 내에서 특정한 패턴을 학습하는 단순 문제에 적용가능하다.

인공지능 분야에 지속적인 연구에 따라, 복잡한 문제를 단일 모델 내에서 좋은 성능과 합리적인 연산량으로 해결하기 위한 연구가 활발히 진행되고 있다. 특히 메모리 요소를 활용한 메모리-증대 신경망은 기존 신경망 모델의 한계적인 장기 기억 의존성 문제에서 벗어나 순환 신경망만으로 다루지 못한 문제들을 해결하고 있다. 또한 메모리 요소의 확장, 연산 효율 개선, 다양한 학습 방법에 대한 연구 등 향후 인공지능 분야의 또 다른 기반이 될 가능성이 높다.

따라서 본 논문에서는 메모리-증대 신경망의 최근 연구 동향을 살펴보고자 한다. 메모리-증대 신경망의 분류가 되는 메모리 네트워크, 뉴럴 튜링 머신, 그리고 스택-증대 신경망의 연구에 대해 먼저 상세하게 설명하고, 각 분류에서는 어떠한 관점에서 차별점을 두고 추가적인 연구가 진행되었는지 분석한다. 또한 각각의 연구에 대한 특징을 요약하고 더불어 문제점들을 설명하며, 이에 따라 향후 연구의 발전 방향을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 메모리-증대 신경망의 세 분류에 대해 상세하게 설명한다. 3장과 4장에서는 각각 메모리 네트워크, 뉴럴 튜링 머신을 기반으로 확장된 연구들을 관점에 따라 분류하여 설명한다. 5장에서 현재까지 연구된 메모리-증대 신경망을 요약하고, 문제점과 연구의 확장 가능 요소에 대해서 토의한다. 마지막으로 6장에서 메모리-증대 신경망의 추후 연구 전망에 대한 결론을 제시한다.

2. 메모리-증대 신경망

2.1 메모리 네트워크

메모리 네트워크[3]는 자연어처리 태스크 중에서 질의-응답 문제를 해결하고자 고안된 네트워크 학습 모델이다. 순환 신경망 기반의 단순 학습 모델은 입력 순차 데이터를 삽입하고 그 다음 예측된 값을 도출하는 방식이기 때문에 장기 기억을 요구하는 질의-응답 문제 해결에는 어려움이 있다. 이는 질의-응답 문제를 해결할 때 필요한 근거(Facts)들을 기억에 유지하고 이를 바탕으로 적절한 답 추론의 한계를 말한다. 반면, 메모리 네트워크는 입력 문장(Input sentences)들을 장기간 저장할 수 있는 메모리 요소에 저장한 뒤, 답을 추론할 때 학습 모델과 상호 작용할 수 있도록 구조화 했다.

Fig. 1은 메모리 네트워크의 전체적인 구조로서, 4개의 요소 I(Input), G(Generalization), O(Output), R(Response)과 메모리 요소로 구성되어 있다.

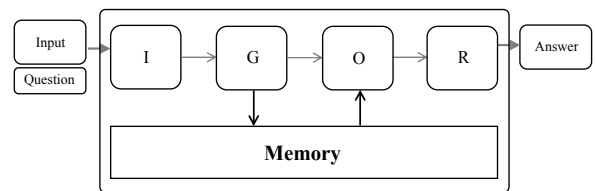


Fig. 1. Memory Networks

- I요소는 입력 값을 받아서 내부 특징 표현형(Internal feature representation)으로 변환시킨다. I요소로의 입력 값은 입력 문장과 질문 문장(Question sentence)이 될 수 있고, 내부 특징 표현형은 입력 값으로부터 인코딩 된 하나의 특징 벡터(Feature vector)로 표현 가능하다.
- G요소는 I요소 결과 값인 내부 특징 표현형을 메모리 요소에 저장한다.
- O요소는 질문 문장이 입력되었을 때 메모리 요소에 저장된 각각의 입력 문장에 Argmax함수를 사용하여 답에 직접적인 근거(Supporting facts)로 적합한 지 스코어링(Scoring)한다. 적합한 근거 단어 혹은 문장에 높은 스코어링 결과를 보인다. 만약 찾는 근거의 개수를 조정하면 스코어링 횟수가 변하여 답 추론 능력과 연산량에 영향을 받게 된다.

- R 요소는 O 요소로부터 얻은 결과를 한 단어 혹은 문장의 형태로 출력하기 위해 사용한다.

각 요소는 순환 신경망 기반 모델을 사용할 수 있기 때문에 MemNN(Memory Neural Networks)이라 칭하기도 한다.

메모리 네트워크의 메모리 요소는 문장 단위로 저장하므로, I 요소의 입력 값은 각 문장에 단어 집합(Bag-of-Words)을 적용한 벡터를 사용한다. 하지만 입력 값이 연속적인 단어인 경우 입력 순차 데이터를 문장 단위로 나눈다. 이 추가적인 과정은 입력 순차 데이터에 구분화(Segmentation) 함수를 적용하여 문장마다의 중단점(Breakpoint)을 지정하도록 학습한다. 벡터화된 문장들은 임베딩(Embedding)을 적용하여 메모리 요소에 메모리 벡터 형태로 저장된다.

메모리 네트워크의 학습은 입력 문장들과 질문 문장의 입력에 대해, 답과 답에 직접적인 근거를 학습 시에 제공하는 강한 지도 학습(Strongly supervised learning)을 사용했다. 그리고 테스트 시에 입력 문장들과 질문 문장을 입력하고 답에 직접적인 근거 제공 없이 답을 출력할 수 있는지를 평가하였다.

메모리 네트워크의 질의-응답 문제 성능을 평가하기 위해서 세 가지 속성(Triples)의 1) 주어(Subject), 2) 관계(Relationship), 3) 목적어(Object) 구조의 문장들로 이루어진 대규모 질의-응답(Large-scale QA)[7] 데이터셋에 대해 실험했다. 이와 더불어 단순한 문법과 상황을 갖는 가상 세계 질의-응답 문제(Simulated world QA)[8]를 실험했다. 가상 세계에서는 4명의 인물, 3개의 사물, 5개의 방으로 이루어져 있고, 이 안에서 인물의 움직임이나 사물을 들고 내려 두는 행동을 문장 형식으로 생성한다. 가상 세계 질의-응답 문제에 대한 예시는 Fig. 2와 같다. Fig. 2에서 답에 직접적인 근거는 해당 문제 번호와 같은 색상으로 표현했다.

Mary took the football. Mary went back to the hallway.
 Bill went back to the kitchen. Mary got the milk there.
 Fred went back to the hallway. Mary left the football there.
 Jeff travelled to the bedroom. Mary passed the milk to Fred.
 Bill journeyed to the bathroom.

(1)What did Mary give to Fred? Milk
 (2)Where is the football now? Hallway

Fig. 2. Example of Simulated World QA

메모리 네트워크는 둘 이상의 문장으로부터 답을 추론하는 난이도 높은 문제에서 기존의 신경망 모델들보다 훨씬 뛰어난 성능을 보였다. 또한 학습에서 보인 적 없는 단어, 해싱을 통한 메모리 사용, 메모리 요소에 쓰여진 시간 정보에 대해 고려함으로써 메모리 네트워크의 성능과 효율에 대한 개선을 시도했다.

하지만 초기 메모리 네트워크는 두 가지 한계점이 있었다.

- 1) 학습 시에 Argmax 함수를 사용하는 하드 주의 기제(Hard attention)를 채택하여 연속적(Continuous)이지 않기 때문에 미분불가능(Non-differentiable)하다는 특징을 가진다. 그로 인해 신경망 학습의 가장 보편적인 방법인 오류 역전파(Back-propagation) 알고리즘을 통한 학습이 불가능하다. 따라서 마진 랭킹 손실(Margin ranking loss)과 같은 추가적인 기법을 사용해야하는 어려움이 있다.
- 2) 강한 지도 학습 방식을 사용한다. 이는 곧 질문 - 답에 직접적인 근거 - 답 셋 사이의 관계에 대한 학습이기 때문에 모델이 질문 문장에 대해서 답을 직접 추론하는 과정이라고 보기 어렵다. 반면 답에 직접적인 근거를 학습 시에 제공하지 않는 약한 지도 학습(Weakly supervised learning)을 적용했을 때는 좋은 성능을 보이지 못했다. 이에 따라 해결 가능한 질의-응답 문제의 범위가 제한적이라는 큰 한계가 있다.

메모리 네트워크는 메모리 요소를 통해 장기 기억을 구현하고, 질의-응답 문제에 적용할 수 있는 새로운 패러다임을 제안하였다. 또한 새로운 모델에 대한 틀을 제안함으로써 발전 가능성을 열어 둔 연구라 할 수 있다. 이후에는 메모리 네트워크 모델을 기반으로 한 확장 연구들이 진행되었다. 이는 궁극적으로 인간에 가까운 질의-응답 문제 해결 능력을 단일 모델로 학습하기 위한 지속적인 연구로 볼 수 있다.

2.2 뉴럴 튜링 머신

뉴럴 튜링 머신[4]은 앨런 튜링(Alan Turing)이 고안한 튜링 머신의 기본적인 구조와 신경망을 활용한 기계 모델이다. 순환 신경망은 이미 튜링-완전(Turing-Complete)하기 때문에 이를 튜링 머신과 결합하여 튜링 머신의 연산 능력을 가질 수 있음을 보이려 했다. 튜링 머신은 값을 저장할 수 있는 무한 길이의 테이프(Tape)와, 이 테이프의 위치를 좌우로 움직이고 읽거나 쓸 수 있는 읽기/쓰기 헤드(Read/Write heads), 그리고 현재 프로그램의 상태(State)에 따라 테이프에 적용할 동작을 헤드가 하도록 결정하는 제어 유닛(Control Unit)으로 이루어져 있다.

Fig. 3은 이러한 튜링 머신 구조를 바탕으로 한 뉴럴 튜링 머신의 구조이다.

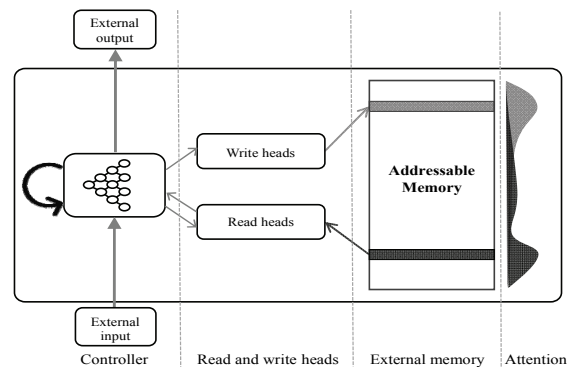


Fig. 3. Neural Turing Machines

- 제어기(Controller)는 신경망을 적용하여 어떠한 특정 메모리 위치에 값을 읽고 쓸 것인지에 대한 가중치를 결정한다.
- 읽기/쓰기 헤드는 메모리의 특정 위치에 읽기/쓰기 연산을 수행한다.
- 외부 메모리(External memory)는 튜링 머신에서 사용되는 테이프처럼 데이터를 저장할 수 있는 무한한 공간이며, 주어진 정보를 담고 있는 벡터들의 집합으로써 하나의 행렬 형태이다. 특히 메모리 네트워크와 달리, 특정 메모리 위치를 선택하여 읽기/쓰기 연산이 가능하기 때문에 주소지정 가능한 메모리(Addressable memory)라는 특징을 갖는다.
- 뉴럴 튜링 머신은 소프트 주의 기제를 사용하기 때문에 메모리 벡터 값이 연속적이며, 주의 기제가 가장 높은 곳으로 읽기/쓰기 연산이 이루어진다.

뉴럴 튜링 머신은 특정 메모리 위치를 선택하도록 소프트 주의 기제를 적용하기 때문에 이를 선택적 소프트 주의 기제(Selective soft attention)이라고도 부른다. 또한, 미분가능한(Differentiable) 연산이 사용되기 때문에 오류 역전파와 알고리즘을 통한 학습이 가능하다는 이점이 있다.

뉴럴 튜링 머신의 주요 기능은 둘로 나누어 볼 수 있다. 1) 메모리로의 읽기/쓰기 연산과 2) 주소지정 메커니즘(Addressing mechanism)이다. 읽기/쓰기 연산은 특정한 메모리 위치로의 주의 기제에 의해 수행 된다. 이러한 주의 기제 정도를 나타내는 가중치(Weight)는 제어기에서 생성되며, 이 가중치를 만들어내는 과정을 곧 주소지정 메커니즘이라고 할 수 있다.

읽기 연산은 제어기에 의해 생성된 가중치와 이전 메모리와의 연산을 통해서 이루어진다. 쓰기 연산은 기존 메모리의 특정 위치에 저장된 값을 삭제 후 새로운 값을 추가하는 두 단계로 나눌 수 있다. 먼저 삭제 벡터(Erase vector)를 통해 이전의 메모리로부터 특정 메모리 위치의 정보를 없애고, 그 다음 추가 벡터(Add vector)를 사용하여 메모리에 새로운 값을 추가한다. 이때 읽기와 쓰기의 벡터 연산에 사용되는 가중치는 같다.

뉴럴 튜링 머신의 주소지정 메커니즘은 메모리 위치에 있는 실제 값과의 유사도를 비교하는 내용기반(Content-based) 주소지정과, 메모리 위치를 기반으로 하는 위치기반(Location-based) 주소지정 방식 두 가지를 사용한다. 한 시간 스텝에서 최종 가중치를 생성하기 위해 하나의 내용기반 주소지정 메커니즘과 세 가지 위치기반 주소지정 메커니즘을 순차적으로 적용한다.

Fig. 4는 뉴럴 튜링 머신에 사용되는 주소지정 메커니즘을 순차적으로 표현한 그림이다.

- 내용기반 주소지정은 코사인 유사도(Cosine similarity)를 통해 특정 메모리 위치의 값과의 유사도를 비교한다.

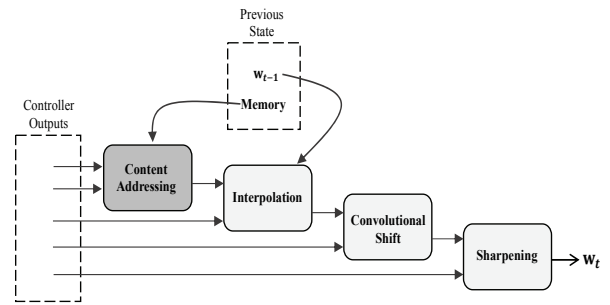


Fig. 4. Addressing Mechanism of Neural Turing Machines

- 위치기반 주소지정 메커니즘은 가장 먼저 내용기반 주소지정의 결과와 이전 시간 스텝에서의 가중치를 게이트를 통해서 적절하게 혼합하는 보간법(Interpolation)을 사용한다. 따라서 내용기반 주소지정 결과를 모두 사용하거나, 반대로 이전 시간 스텝의 가중치를 모두 사용할 수 있다.
- 다음으로 보간법의 결과에 컨볼루션 이동(Convolutional shift)을 적용한다. 이는 특정 메모리 위치를 기준으로, 이전 위치 혹은 현재 혹은 이후 위치로의 회전을 의미한다. 이때 메모리 위치의 전체 값이 회전하는 것이 아니라 회전의 정도를 실수 값에 의해 분산할 수 있다는 특징이 있다. 이렇게 하나의 값이 여러 곳으로 분산되는 것을 번짐(Blur)이라고 표현한다.
- 번짐에 의해 분산된 컨볼루션 이동의 결과 값을 선명화(Sharpening)를 적용하여 값을 뚜렷하게 결정한 최종적인 가중치를 만든다.

내용기반 주소지정은 특정한 메모리 위치로의 점프 이동을 목적으로 사용하는 반면, 위치기반 주소지정은 메모리 위치의 반복 사용과 메모리 위치의 이전 혹은 이후로의 이동이 가능하게 한다.

뉴럴 튜링 머신은 지도 학습(Supervised learning)으로 학습하여 제어기의 메모리 위치 선택 능력과 알고리즘적 문제(Algorithmic tasks)에 대한 학습 능력을 보였다. 뉴럴 튜링 머신의 구현에 필요한 메모리의 크기, 읽기/쓰기 헤드의 개수, 헤드의 이동 범위 등 여러 인자들은 비교적 자유롭다. 하지만 실제로 제어기에 어떤 순환 신경망을 사용하는가에 따라서 성능 차이가 발생할 수 있다. 따라서 성능 비교에는 장단기 기억 모델, 장단기 기억 모델을 제어기로 사용한 뉴럴 튜링 머신, 피드포워드(Feedforward) 제어기를 사용하는 뉴럴 튜링 머신이 사용되었다. 실험 태스크에 사용되는 입력 값은 5-8 bits의 이진 벡터이며, 입력이 끝난 후 출력이 진행될 때 새로운 입력이 주어지지 않는다. 또한 태스크의 특성에 따라 여러 개의 읽기 헤드를 병렬적으로 사용할 수 있다.

실험 태스크는 총 5가지이다. 먼저 장단기 기억 모델과 비교하여 장기 기억 보존성을 잘 보인 태스크는 입력 순차 데이터를 그대로 출력해내는 복사(Copy), 입력 순차 데이터와

스칼라 값의 반복 횟수를 입력했을 때 반복 횟수만큼 입력 데이터를 출력하는 반복 복사(Repeat copy)이다. 복사는 반복문(Loop)을 수행하는 알고리즘 학습, 반복 복사는 중첩 반복(Nested loop)을 수행하는 알고리즘을 학습하는 것이라 볼 수 있다. 이외에, 구분자로 나누어진 여러 순차 데이터들을 입력한 후 그 중 특정 순차 데이터를 입력했을 때 다음에 순서에 나타날 데이터를 찾는 연상 기억(Associative recall), 입력 순차 데이터의 분포를 통해 생성된 데이터들의 패턴을 인지하여 다음 값을 예측하는 동적 엔그램(Dynamic N-Grams), 그리고 임의의 이진 벡터와 스칼라 우선순위 값을 입력하였을 때 우선순위에 따라 정렬하는 순차 정렬(Priority Sort)에서 문제 해결능력을 보였다. 실험된 모든 태스크에서 뉴럴 튜링 머신을 사용할 때 장단기 기억 모델을 사용할 때보다 일관적으로 학습 비용이 더 적고 좋은 성능을 보였다.

뉴럴 튜링 머신은 현대 컴퓨터의 기초 구조라 할 수 있는 튜링 머신과 인공 신경망을 융합하여, 메모리와의 상호 작용을 제어할 수 있는 메커니즘을 학습하는 모델이다. 이때 사용되는 주소지정 가능한 메모리와 주소지정 메커니즘은 연산량과 성능에 가장 큰 영향을 주는 핵심 요소가 된다. π -라서 뉴럴 튜링 머신의 주소지정 메커니즘은 추후 연구가 필요한 주요 요소가 되었다. 뉴럴 튜링 머신은 주소지정 메커니즘에 내용기반 주소지정과 위치기반 주소지정을 모두 사용하기 때문에 학습에 필요한 파라미터의 수가 많고 실제 학습 과정에서도 어려움이 있다. 이후에 주소지정 메커니즘을 개선하기 위한 연구들이 진행되었으며, 강화 학습(Reinforcement learning)을 적용한 연구와 복잡도를 줄이기 위한 연구가 진행되었다.

2.3 스택-중대 신경망

스택-중대 신경망은 메모리-중대 신경망의 한 분류로서, 메모리 네트워크나 뉴럴 튜링 머신과는 달리 구조화된 메모리 형태를 활용한다. 구조화된 메모리는 컴퓨터 과학 분야에서 제시하는 자료구조 스택, 큐, 텍, 그리고 리스트를 활용하며, 구조화된 무한한 크기의 메모리를 순환 신경망 구조에 접합시킬 수 있다.

스택-중대 신경망은 입력 순차 데이터 내에서 패턴을 학습하는 단순 알고리즘 문제 해결을 학습하는 것에 사용된다. 최근 스택-중대 신경망과 관련된 논문[5, 6]은 푸시다운 오토마타(Pushdown Automaton)의 구성 요소인 외부 스택 메모리(External stack memory)와 순환 신경망을 연결 지었던 연구[9, 10]의 후속 연구라고 할 수 있다.

스택-중대 순환망 연구[5]는 심층 신경망에 스택 혹은 리스트 구조의 메모리 요소를 활용한 모델을 제안하였다. 구조화된 메모리 요소는 비지도 학습(Unsupervised learning)을 적용한 단순 순서 예측 문제(Simple sequence prediction problems)와 지도 학습을 적용한 이진 덧셈(Binary addition)

Table 1. The Examples of Simple Sequence Prediction

	Generated sequence	Example
1	$\{a^n b^n n > 0\}$	aaabbbba ba abba aabbb
2	$\{a^n b^n c^n n > 0\}$	aabbccca aabbbccca bc

문제와 같은 알고리즘적 패턴(Algorithmic patterns)을 학습하는 것에 활용된다. 두 태스크의 특성상 다중 스택 구조를 사용하는 실험들을 주로 수행하였다. 단순 순서 예측 문제 예시는 Table 1과 같다.

Table 1에서 주어진 순차 데이터가 있을 때 검정 색으로 표현된 순차 데이터를 입력하면, 주어진 순차 데이터의 패턴을 학습하여 빨간 색으로 표현된 순차 데이터가 출력하는 방식이다.

따라서 스택-중대 순환망은 학습을 통해 스택에 데이터를 넣는 푸시(Push) 연산, 데이터를 빼내는 팝(Pop) 연산, 그리고 아무 연산도 수행하지 않는 연산 없음(No-op)을 선택하게 되고, 이를 바탕으로 기존의 심층 신경망이 찾아내지 못했던 알고리즘적 패턴을 학습할 수 있다. 이는 구조화된 메모리 요소의 활용을 통해 신경망의 보편성을 확보하기 위한 초석이 되었다.

뉴럴 스택(Neural Stack)[6]은 스택-중대 순환망과 마찬가지로 구조화된 무한의 메모리를 순환 신경망과 접합하고자 했다. Fig. 5는 순환 신경망과 뉴럴 스택이 상호작용하는 모델의 구조이다.

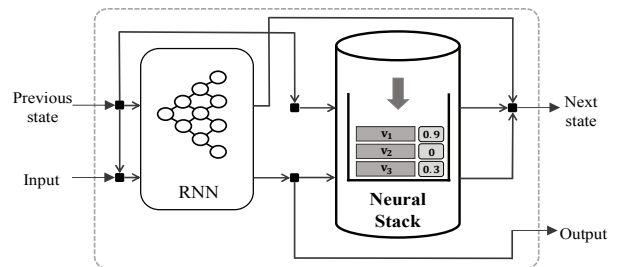


Fig. 5. Neural Stack

뉴럴 스택의 전체적인 구조는 제어기의 역할을 하는 신경망과 메모리 요소의 역할을 하는 뉴럴 스택이 연결되어 있으며, 스택 구조의 메모리에 얼마나 데이터를 푸시하고 팝할지 신경망을 통해 학습하는 구조이다. 이때 [5]에서 사용된 스택과 달리 연속적 스택(Continuous stack)을 사용하여 미분가능하다는 특징이 있다. 연속적 스택은 부분적인 푸시/팝 연산이 가능하며, 저장했던 데이터가 스택에 유지되어 있는 정도로 생각할 수 있다. Fig. 5는 저장된 데이터 벡터 v 와 그 값이 연속적 스택에 유지되어 있는 정도를 표현했다. [6]은 뉴럴 스택과 뉴럴 큐, 뉴럴 텍을 구조화된 메모리로 사용하였으며, 트랜스덕션 태스크(Transduction tasks)에서 자료

Table 2. Summary of Stack-Augmented Neural Networks

Model	Data structure	Tasks	Characteristics
Stack-Augmented Recurrent Nets[5]	Stack, Doubly linked list	Simple sequence prediction problems, Language modeling	Capacity to count and to memorize sequences via trainable memory
Neural Stack[6]	Stack, Queue, Deque	Synthetic transduction tasks, ITG(Inversion Transduction Grammars) transduction tasks	Continuously differentiable of data structures, Logically unbounded memory, Taking constant time to push/pop

구조에 따른 성능 결과를 보였다. 트랜스덕션 태스크란 기계 번역(Machine Translation)[11]처럼 한 문장을 다른 문장으로 변환시키는 것을 학습하는 태스크를 일컫는다. 따라서 입력 순차 데이터를 순서에 맞게 출력하는 순차 복사(Sequence copying), 입력 순차 데이터를 역순서로 출력하는 시퀀스 역전(Sequence reversal), 그리고 짝수의 심볼을 갖는 입력 순차 데이터에서 각각의 두 심볼 단위를 교체(Swap)하는 바이그램 플리핑(Bigram flipping)과 같은 합성 트랜스덕션 태스크(Synthetic transduction tasks)에서 기존의 장단기 기억 모델보다 좋은 성능을 보였다. 예를 들어 시퀀스 역전 태스크에서 뉴럴 스택을 사용하게 되면, 스택 자료 구조가 선입력 후출력 구조이기때문에 데이터 입력 시 순차적인 푸시 연산과 출력 시 순차적인 팝 연산을 수행하도록 학습시킨다.

이외에 주어-동사-목적어 구조를 주어-목적어-동사 구조로 변환하는 SVO to SOV 태스크, 성별에 맞는 문법으로 변환하는 성별 동사 변형(Gender conjugation) 태스크와 같이 언어 번역에 사용될 수 있는 ITG(Inversion Transduction Grammar) 트랜스덕션 태스크에서도 좋은 성능을 보였다. 이처럼 뉴럴 스택은 순환 신경망과 뉴럴 튜링 머신 사이의 중간 모델로, 뉴럴 튜링 머신에 비해 비교적 단순한 알고리즘 문제 해결에 적합한 단순 구조를 큰 이점으로 제시할 수 있다.

Table 2는 스택-증대 신경망과 관련된 스택-증대 순환망, 뉴럴 스택의 요약 및 비교 표이다.

스택-증대 신경망과 메모리-증대 신경망의 명확한 차이점은 근본적으로 메모리 구조의 차이에서 발생한다. 뉴럴 스택은 주소지정 메커니즘을 적용하는 메모리-증대 신경망과 달리, 필요한 메모리 위치를 찾을 필요 없이 메모리 순서가 정해져 있어 연산이 훨씬 간단하다.

스택-증대 신경망 기반 모델은 태스크에 적합한 자료구조를 사용했을 때 메모리-증대 신경망보다 훨씬 간단한 연산으로 충분히 비슷한 성능을 보이는 효율적인 측면이 있다. 하지만 사용되는 자료구조가 태스크 성능에 큰 영향을 주는 한계(Task-specific)가 있다. 예를 들어 시퀀스 복사 태스크는 뉴럴 큐를 사용할 때 최적의 성능을 보이고, 시퀀스 역전은 뉴럴 스택을 사용할 때 최적의 성능을 보인다. 이러한 자료 구조 형태의 메모리 구조는 적용 범위가 제한적이라는 한계점을 갖는다.

3. 메모리 네트워크 확장 연구

3.1 종단간 메모리 네트워크

종단간 메모리 네트워크(End to End Memory Networks)[12]는 기존 메모리 네트워크의 하드 주의 기제와 강한 지도 학습을 사용하는 데에 발생하는 두 가지 문제점을 해결하기 위해 제안된 모델이다.

Fig. 6은 종단간 메모리 네트워크의 구조이다.

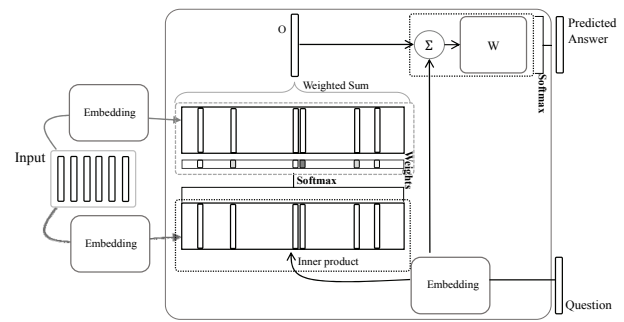


Fig. 6. End-to-End Memory Networks

- 입력 문장들은 단어 집합으로 문장 단위로 인코딩하고 이에 임베딩 행렬을 적용하여 메모리 벡터로 저장한다.
- 질문 문장 또한 인코딩하고, 질문 문장 벡터 값은 답을 추론하기 위한 메모리 벡터와의 연산과 최종 답을 결정하기 위한 연산에 사용된다.
- 답 추론 과정은 메모리 벡터들과 질문 문장의 소프트 주의 기제를 통해 질문과 관련 깊은 메모리 벡터에 집중한다. 다시 이 결과를 입력 문장 메모리 벡터와 가중 합(Weighted sum)하여 결과를 도출한다.

종단간 메모리 네트워크는 Fig. 6의 단일 층 구조를 반복하여 Fig. 7과 같은 다수의 층 구조로 사용된다. 이를 다중 홉(Multiple hops)구조라 칭하며, 답에 직접적인 근거를 추론하기에 적합한 구조이다.

종단간 메모리 네트워크는 초기 메모리 네트워크의 Argmax함수를 사용한 하드 주의 기제 대신에 Softmax함수를 사용한 소프트 주의 기제(Soft attention)를 사용하였다. 이러한 변화는 단일 모델을 통해 직접적으로 정답을 도출할 수 있는 종단간 모델을 구현하여 실용성을 대폭 증가시켰다.

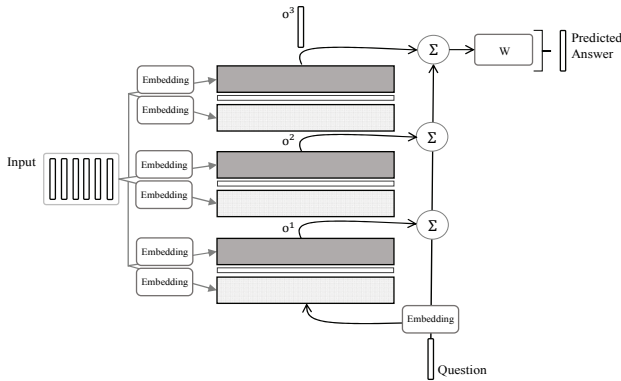


Fig. 7. Multiple hops of End-to-End Memory Networks

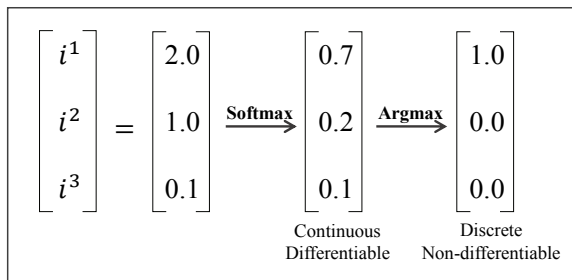


Fig. 8. An example of Softmax and Argmax function

초기 메모리 네트워크에서 사용했던 Argmax함수와 종단간 메모리 네트워크의 Softmax함수 각각의 적용에 따른 예시는 Fig. 8과 같다.

또한 종단간 메모리 네트워크는 다중 홉 구조를 통해 강한 지도 학습의 한계를 극복하고자 했다. 그에 따라 답에 직접적인 근거 없이 학습하는 약한 지도 학습에서도 강한 지도 학습에 못지않은 성능 결과를 보였다는 점이 큰 기여이다.

3.2 메모리 네트워크의 활용 및 확장성

메모리 네트워크에 대한 연구를 바탕으로, 대화 시스템(Dialog Systems)에 모델을 활용하거나[13-15] 질의-응답 문제의 적용 범위를 확장한 연구가 진행되었다[16-19].

대화 시스템은 대화 속에서 질의와 응답에 해당하는 문장을 포함할 수 있기 때문에 질의-응답 문제 해결을 위한 메모리 네트워크의 활용 측면으로 생각할 수 있다. 따라서 대화 시스템을 수행하는 것에 목적을 두고 종단간 메모리 네트워크 구조를 적용한 연구가 진행되었다.

종단간 대화 시스템(End-to-End Dialog Systems)[13]은 종단간 메모리 네트워크를 활용하여 대화 시스템을 구조를 구현하고자 했다. 종단간 메모리 네트워크에서 일반적인 지식기반의 정보를 장기기억 메모리로, 대화를 통해 만들어지는 문맥을 단기기억 메모리로 분류하였다. 특히, 지식기반의 정보를 저장한 장기기억 메모리를 찾는 과정에서 해시 탐색(Hash lookup)을 사용하였다.

대화기반의 언어 학습(Dialog-based learning)[14] 연구는 인간이 대화를 통해 새로운 정보를 학습할 수 있다는 아이디어를 착안하여, 대화기반의 언어 학습 방법론을 제안하였다. 마찬가지로 종단간 메모리 네트워크의 구조를 기반으로 하여, 기존 구조에서 층을 분리하는 등 구조의 변형을 통해 대화기반의 지도 태스크(Dialog-based supervision tasks) 수행 능력을 실험하였다.

종단간 메모리 네트워크 모델을 목표 지향 대화 시스템(Goal-oriented Dialog systems)에 활용한 연구[15]도 진행되었다. 목표 지향 대화는 단순히 대화에서 문맥에 알맞은 말을 생성해내는 것뿐만 아니라, 사전에 정의된 특정한 목표를 달성하기 위한 대화를 일컫는다. 목표 지향 대화를 수행할 수 있는 모델은 충분히 실용적인 대화 시스템으로 적용 가능하지만, 특정한 도메인의 정보를 많이 필요로 하기 때문에 제한된 도메인에서의 수행가능성을 보이고자 했다.

[15]는 종단간 메모리 네트워크 모델을 활용하여 식당 예약과 관련된 종단간 대화 시스템에 적용했다. 식당 예약에 필요한 식당 위치, 연락처, 음식 종류, 수용 가능 인원 등 식당과 관련된 정보는 지식베이스(Knowledge Base)에 저장되어 있고, 식당 정보 7가지를 묶은 지식베이스 개체(KB entity) 형태로 사용한다.

목표 지향 대화 시스템은 5가지 태스크를 절차적으로 수행한다. 1) 태스크1은 식당 예약에 필요한 정보를 사용자와 대화 봇(Bot)의 대화로부터 추출하여, 지식베이스 개체의 정보 중 해당하는 일부분으로 API 요청을 생성한다. 2) 태스크2는 이후 대화에서 예약 정보의 수정이 있을 경우 해당 정보를 반영하여 API 요청을 업데이트 한다. 3) 태스크3는 API 요청에 포함된 정보를 기반으로, 해당하는 식당들의 나머지 정보들을 지식베이스에서 개체 형태로 추출하여 목록화하고, 그 중 식당을 사용자에게 추천한다. 4) 태스크3에서 사용자가 예약하려는 식당을 선정했다고 가정했을 때, 태스크4는 이후 사용자에게 의한 식당 추가 정보 요청에 대해 지식베이스 개체로부터 정보를 제공한다. 5) 마지막으로 태스크5는 태스크1~4의 전체 대화를 수행해내는 것을 말한다. [15]는 모든 태스크에 각각의 종단간 메모리 네트워크를 적용하고 다중 홉을 통해 적절한 답변을 생성하도록 구조화 하였다.

목표 지향 대화 시스템에 사용된 종단간 메모리 네트워크는 식당 예약 대화에서 세 가지 과정을 고려한다. 1) 저장 및 표현: 어떻게 대화 기록을 표현하고 이를 메모리에 저장할 지 고려한다. 시간 순서로 사용자의 발언(User utterance)과 그에 대한 봇의 답변이 주어질 때, 모든 발언과 답변은 문장 단위의 단어 집합으로 인코딩 된다. 시간 스텝 t 에서 메모리에 저장되는 벡터는 시간 스텝 t 까지의 사용자의 모든 발언과 $t-1$ 까지의 봇의 답변이다. 시간 스텝 t 에서의 봇의 답변은 시간 스텝 t 에 대한 종단간 메모리 네트워크의 최종 산출 결과이다. 2) 주의 기제: 응답에 대한 추론을 하기 위해 어떤

메모리로부터 읽어올 지 고려한다. 이때 시간 스텝 t 의 사용자 발언과 각 메모리 벡터와의 소프트 주의 기제를 통해 읽어올 메모리 후보들을 선정하게 된다. 3) 응답 선택: 어떤 답을 출력할 것인가를 고려한다. 생성된 모든 후보 중 최종적으로 예측된 응답을 선택한다.

규칙기반 시스템(Rule-based system), 정보 검색(Information retrieval) 모델, 지도 임베딩(Supervised embedding) 모델과의 성능 비교 결과, 종단간 메모리 네트워크를 사용했을 때 가장 좋은 성능을 보였다. 이처럼 메모리 네트워크 기반 모델을 대화 시스템에서 활용함으로써 기존 모델보다 향상된 결과를 꾀하는 연구들이 진행되었다.

메모리 네트워크 모델을 활용하여 질의-응답 문제의 적용 범위를 확장시킨 연구들도 있었다. [16]은 방대한 지식 베이스(Large-scale Knowledge Base) 기반의 질의-응답 문제를 최적화된 메모리 네트워크 구조에서 해결하고자 하였다. 메모리 네트워크는 주어진 문맥에서 질문에 해당하는 근거들을 찾아 답을 추론하지만 대규모의 메모리에서 원하는 답을 찾는 것은 쉽지 않다. 따라서 지식 베이스 기반의 방대한 질의-응답 문제를 생성하고, 대규모의 메모리에서 단일 근거(Single fact)를 찾아내는 문제에 초점을 두었다. 이를 위해 방대한 지식 베이스를 메모리에 저장하는 전처리 과정과 저장된 메모리로부터 후보군을 추출하는 과정이 지식 베이스에 특화하였다. 각각의 문장들은 지식 베이스에서 세 가지 속성(주어, 관계, 목적어)으로 표현한다. 주어와 목적어는 개체(Entity)에 속하고, 관계들은 개체 간의 연결 고리이다. 질문 문장은 엔그램 집합(Bag-of-Ngrams)을 사용하며, 세 가지 속성에서 목적어를 비운 형태로 표현 된다. 질문 문장은 지식 베이스의 데이터들과 유사도를 측정하여 정답 후보군을 추론하며, 추출된 정답 후보군과 질문 문장의 사인 유사도를 계산하여 최종 정답 개체를 출력한다.

[17]은 인간과 대화가능한 수준의 질의-응답 문제 적용 범위로 확장하기 위해, 메모리 네트워크를 세 가지 측면의 연구 방향을 제시하였다. 1) 적응적인 메모리(Adaptive memory)를 활용하여 모델에서 스코어링하는 답의 근거 개수를 2개로 제한하지 않도록 하고, 출력 값 또한 다수의 단어를 사용할 수 있도록 하였다. 2) 3-그램 집합(Bag of 3-grams)의 사용을 제안하였다. 3) 매칭 함수에 비선형성을 갖는 하이퍼볼릭 탄젠트(tanh)함수를 활용하여, 단어 간의 순서가 주요 추론 요소가 되는 태스크의 성능을 높였다.

메모리 표현법의 확장을 제안한 연구[18]는 종단간 메모리 네트워크 구조를 사용하여 동화책의 내용처럼 긴 입력 문장에서 빈칸 내용을 추론하는 것을 목표로 한다. 빈칸 문장의 근처 단어로부터 정답을 유추하는 지역적 문맥(Local context)이나 전체적 문맥(Wider context)을 이해하기 위해 세 가지 메모리 인코딩 방법을 제안하고 각각의 문제 해결 능력을 비교하였다. 1) 각 단어를 독립적인 메모리로 표현하

고 저장하는 방법으로, 가장 낮은 수준의 메모리 표현법이다. 2) 슬라이딩 윈도우를 바탕으로 문장을 몇 개의 구절로 표현하여 각각을 메모리로 저장하였다. 3) 문장 전체를 인코딩하여 메모리 벡터로 저장하였다. 각각의 메모리 표현법을 사용하여 종단간 메모리 네트워크를 훈련한 결과, 슬라이딩 윈도우 기반의 구절 단위의 메모리 표현법을 적용한 모델이 지역적 문맥과 전체적 문맥에 대해 가장 정확한 결과를 도출하는 결과를 보였다. 이는 윈도우 기반의 메모리 표현법을 통해 모델이 문맥을 이해하는 기능이 대폭 유연해짐을 나타낸다. 또한 구절 단위의 메모리 벡터를 모두 메모리에 저장하여 훈련한 모델 외에, 필요한 윈도우 메모리를 선택하는 효율적 방안을 적용하였다.

또한 위의 연구에서 지식 베이스 기반의 질의가 갖는 한계점을 극복하기 위해, 지식베이스와 더불어 위키피디아(Wikipedia) 같은 원시 문서 모음에서 질의-응답을 해결할 수 있는 키-값 메모리 네트워크(Key-Value Memory Networks)[19]가 제안되었다. 지식 베이스는 다양한 질문에 해당하는 정보를 보유하지 못한다는 필연적인 불완전성과 고정된 스키마로 인해 제한적이다. 결과적으로 제한된 도메인 문제에는 지식 베이스로 충족될 수 있지만, 오픈 도메인에 대한 일반적인 답변은 더 많은 정보를 포함하고 있는 원시 문서 모음으로의 확장이 필수적이다. 제안된 키-값 메모리 네트워크는 종단간 메모리 네트워크의 전체적인 구조와 매우 흡사하다. 하지만 키-값 쌍의 메모리 표현 구조를 바탕으로 질의-응답 문제를 해결하는 과정이 독창적이다. 키-값 쌍의 메모리 표현 구조는 지식 베이스 또는 원시 문서 모음에서의 문맥이 메모리에 저장되는 방식이며, [18]에서 제안한 다양한 메모리 인코딩을 적용하고 결과를 비교 분석하여 최적의 키-값 메모리 구조를 제안하였다.

메모리 네트워크와 관련된 연구들은 질의-응답 문제 해결에 사용된 모델의 성능 비교를 위해 데이터셋을 제안하였다. Table 3은 제안된 데이터셋에 대해 정리한 표이다.

[13]은 영화 도메인의 대화 시스템에 대한 벤치마크를 제안하였다. 단순히 문맥에 알맞은 말을 생성해내는 것이 목적인 기존 벤치마크와는 다르게, 질의-응답 문제, 추천, 질의-응답과 추천의 복합, 그리고 궁극적으로 잡담에 가까운 대화를 할 수 있는 것에 목적을 둔 벤치마크이다. 하지만 도메인이 영화로 한정되어 있기 때문에 일반화된 대화시스템을 평가하기에는 한계가 존재한다.

질의-응답 데이터셋으로 이전에 사용되었던 Web-Questions[20]은 데이터셋의 규모가 상대적으로 작다. 따라서 프리베이스(FreeBase)와 같은 지식 베이스를 기반으로 하여 방대한 양의 데이터들을 포함하는 SimpleQuestions [16]을 제안하고, 이를 통해 대규모 질의-응답 문제 해결 능력을 평가하기 위해 사용되었다.

[17]은 질의-응답 문제와 관련된 20가지 태스크들의 벤치

Table 3. Summary of proposed dataset

Dataset	Description
The Movie Dialog Dataset (MovieQA)[13]	Built from large-scale real-world sources. Test the ability of End-to-End Dialog Systems. Choose the domain of movies and movie related entities.
SimpleQuestions[16]	Built from Knowledge Base Freebase with human generated questions. Format triples as (Subject, Relationship, Object). Rephrase questions as (Subject, Relationship, ?).
bAbI tasks[17]	Generate data in many different scenarios/simulations. Construct 20 kinds of synthetic tasks. Designed to be prerequisites for text understanding and reasoning.
The Children's Book Test(CBT)[18]	Built from books Project Gutenberg. Designed to be measure how models can capture both local and wider linguistic context in children's stories.
WikiMovies[19]	Built from only the QA part of MovieQA Test the ability of models to directly read documents to answer questions. Measure performance of models when knowledge source is switched from a KB to documents

마크인 bAbI 태스크를 제안하였다. bAbI 태스크는 문맥에 대한 모델의 이해도와 추론 능력을 평가한다. 궁극적으로는 실 세계에서 발생할 수 있는 여러 형식의 질의-응답 문제에 대한 성능을 평가하기 위함을 목표로 한다. 따라서 최근 질의-응답과 관련된 많은 연구에서 bAbI 태스크를 주요 성능 평가 지표로 사용하는 경향을 확인할 수 있다.

한편, [18]은 동화책 내용의 일부분을 모델에 입력하고 다음 문장의 빈칸에 알맞은 정답 단어를 추론하는 과정을 위한 일련의 벤치마크(Children's Book Test, CBT)를 제안하였다. 이는 동화책 내용 중 짧은 구간의 문맥과 긴 구간의 문맥을 모두 이해할 수 있는 모델인지 평가하기 위함이다.

3.3 메모리 요소 확장 기법

메모리 요소는 메모리 네트워크 구조의 가장 핵심적인 요소로서, 학습 적용 범위 확장과 더불어 집중적으로 연구되었다. 특히 메모리 네트워크 구조의 메모리 요소를 확장하고 구체화하여, 멀티 모달리티(Multi-modality)를 실현시키거나 bAbI 태스크의 성능이 뛰어난 모델에 대한 연구가 진행되었다.

동적 메모리 네트워크(Dynamic Memory Networks)[21]는 자연어 처리에서 질의-응답 문제뿐만 아니라 품사 태그(POS tagging), 감정 분석(Sentiment analysis)을 포함한 세 가지 문제를 하나의 네트워크 모델을 통해서 해결하고자 했다. 핵심 아이디어는 특정 시간에 대한 일련의 기억을 나타내는 에피소드(Episode)를 만들어 내고 이를 통해 메모리를 업데이트 하여, 동적으로 답에 직접적인 근거들을 찾아내는 것이다. 동적 메모리 네트워크는 각각의 반복(Iteration)을 거쳐서 실제 최종 답을 도출하기까지 여러 근거 문장들을 순차적으로 추론한다. 이때 반복 과정에서 주의 기제가 동적으로 변하여 추론 능력을 갖추므로 동적(Dynamic)이라고 표현한다.

Fig. 9는 동적 메모리 네트워크의 구체적인 구조이다. 동적 메모리 네트워크는 입력(Input), 질문(Question), 에피소딕 메

모리(Episodic memory), 답(Answer) 모듈로 네 가지 모듈로 나눌 수 있다. 각 모듈은 모두 순환 신경망 기반의 학습 모형인 GRU(Gated Recurrent Unit)를 사용한다.

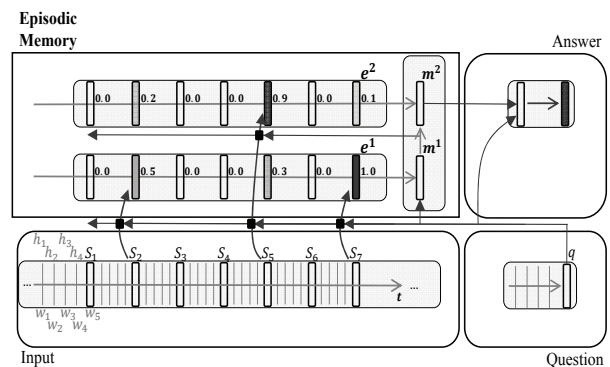


Fig. 9. Dynamic Memory Networks

- 입력 모듈은 입력 문장들에 대해 단어 단위 w 로 임베딩하여 GRU에 순차적으로 입력하게 되고, 문장의 마지막 입력 단어에 대한 출력 값으로 한 문장을 대표하는 하나의 표현 값(Fact representation) S 을 생성해낸다.
- 질문 모듈은 입력 모듈과 같은 방식으로 GRU를 적용하여 한 문장의 질문의 표현 값 q 을 만들어낸다. 이때 입력 단어에 적용되는 임베딩 행렬은 입력 모듈과 같다.
- 에피소딕 메모리 모듈은 세 가지 과정의 반복을 수행한다. 1) 입력 문장 S^i , 질문 문장 q , 이전 메모리 m^i 를 통해 에피소드 e^j 를 생성한다. 첫 반복에서는 이전 메모리가 없기 때문에 질문 값 q 과 입력 값 S^i 와의 관계 정도를 계산한다. 이때 관계 정도에 따라서 주의 기제를 적용하여 첫 입력 문장에서 하나의 에피소드 e^j 를 생성한다. 2) 에피소드 e^j 를 입력 문장의 순서에 따라 주의 기제를 적용하여 업데이트한다. 3) 마지막 입력 문장에 의해 업데이트

된 에피소드 e^j 를 메모리 m^j 을 업데이트하기 위해 사용한다. 이러한 세 가지 과정의 반복을 통해 메모리 m 을 업데이트하는 과정에서 답을 추론할 수 있다. 이를 반복적 주의 기제(Iterative attention)라 한다.

- 답 모듈은 질문 값 q 와 에피소드 메모리 모듈로부터 생성된 최종 메모리 m^f 에 Softmax함수를 적용하여 예측 값을 출력한다.

동적 메모리 네트워크는 앞서 나열한 세 가지 자연어 처리 태스크들에 대해 각각 좋은 성능의 결과를 보였다. 비록 각각의 문제마다 모델의 모듈들을 세부적으로 조정하여 작동시켰지만, 결국 하나의 모델이 세 가지 자연어 처리 문제를 수행할 수 있었다. 이러한 응용적인 측면을 부각시킴으로써 하위 자연어 처리 태스크가 곧 하나의 질의-응답 문제로 치환될 수 있다는 것을 보였다. 또한 핵심이 되는 에피소드 메모리 모듈에 반복 횟수가 성능에 미치는 영향에 대한 추가적인 실험을 수행하여, 반복적 주의 기제의 중요성을 검토하였다.

동적 메모리 네트워크 모델의 학습 부하 면을 살펴봤을 때, 모든 모듈에서 학습 파라미터 수가 장단기 기억 모델보다는 GRU를 사용함으로써 학습에 필요한 부하를 줄이고자 하였다. 하지만 초기 동적 메모리 네트워크는 GRU의 한 방향성만을 갖는 구조이기 때문에 약한 지도 학습을 적용하여 좋은 성능을 도출하기 어려웠다.

동적 메모리 네트워크에 약한 지도 학습을 가능하게 하여 기존의 한계를 벗어나고, 이미지 기반 질의-응답 문제 또한 해결함으로써 멀티 모달리티에 기여한 연구[22]가 추가적으로 진행되었다. 일반적으로 초기 동적 메모리 네트워크 모델을 DMN[21]이라 하며 추가 연구를 DMN+[22]라 칭한다. 해당 연구는 입력 모듈에서 GRU를 적용하는 층을 수정하였다. 기존 DMN 모델의 입력 모듈은 하나의 GRU를 사용했다. 하지만 DMN+ 모델은 두 개의 GRU를 사용하는 입력 퓨전층(Input fusion layer)를 적용하여, 문장 간 정보의 흐름을 전후로 전달할 수 있도록 개선하였다.

순환 개체 네트워크(Recurrent Entity Networks)[23]는 동적 메모리 네트워크와 유사하게 동적으로 메모리로부터 추론하지만, 병렬적인 메모리 요소를 활용한다. 병렬적인 메모리 요소의 구조는 종단간 메모리 네트워크의 다중 홉 혹은 동적 메모리 네트워크의 반복 과정을 대체한다. 병렬적인 메모리 구조에는 다수의 개체(Entity)를 활용하는데, 이때 하나의 개체마다 하나의 GRU를 갖으며, 개체는 사용자에 의해 사전에 정의되거나 학습 시에 정해질 수 있다. 순환 개체 네트워크의 구조는 크게 입력 인코더 모듈, 동적 메모리 모듈, 그리고 출력 모듈로 구성되어 있다. 모델의 핵심인 동적 메모리 모듈은 개체의 변경을 추적해가며 개체의 메모리 값으로 여기는 은닉층(Hidden layer)을 갱신하는 역할(Tracking world state)을 한다. 모든 갱신이 종료되었을 때, 각 개체에 해당하는 최종 메모리 값은 주어진 질문에 대한 정답을 추론하는 데에

사용된다. 순환 개체 네트워크는 질의-응답 문제 추론 능력의 대표적인 평가 지표라 할 수 있는 bAbI 태스크에서 가장 좋은 성능을 보였다.

3.4 구조 변형을 통한 복잡도 개선

방대한 양의 데이터를 저장하기 위한 메모리를 다룰 때 필연적으로 발생하는 연산량을 줄이기 위한 기법을 제안한 다양한 연구들 또한 이루어지고 있다.

키-값 메모리 네트워크[19]는 키-값 메모리 구조를 사용하여 대용량의 데이터에서 키-값 후보군을 추려내는 과정을 통해 연산량을 줄이고자 했다. 키는 주어진 질문과 일치하는 문장을 추려낼 수 있도록 설계되고, 값은 정답을 추론하도록 설계된다. 따라서 키-값 메모리 네트워크는 키와 질문의 소프트 주의 기제 방법으로 질문에서 요구하는 키를 도출하고, 그에 해당하는 값을 읽는 과정을 거쳐 정답을 추론한다. 이때 키-값 구조의 주요 장점인 키를 이용한 해싱(Hashing)을 통해 후보군이 되는 키-값 쌍의 개수를 축소시켜, 대용량 메모리에 필요했던 연산량을 줄일 수 있다.

계층적 메모리 네트워크(Hierarchical Memory Network)[24]는 소프트 주의 기제와 하드 주의 기제 두 가지 방법론을 단계적으로 적용하기 위한 계층적 메모리 구조를 제시하여, 대용량의 메모리를 효율적으로 다루고자 하였다. 소프트 주의 기제는 참조하고자 하는 메모리의 위치를 특정 짓기 위한 행렬 연산을 수행함에 있어 연산 복잡도가 메모리에 저장된 양에 비례한다. 그로 인해 메모리의 크기가 절대적으로 커짐에 따라 학습 및 테스트 시간이 선형적으로 증가할 수 있다. 반면에 하드 주의 기제를 사용하게 되면 성능 향상은 이를 수 있으나 활성화 함수(Activation function)가 미분가능하지 않기 때문에 오류 역전과 알고리즘의 사용이 불가능하다. 그러므로 강화 학습을 이용한 가중치의 보정을 수행해야하기 때문에 학습 결과를 도출하는 데에 어려움이 있다. 따라서 계층적 메모리 네트워크는 세 가지 과정을 적용한다. 1) 기존의 평판 메모리 구조(Flat memory structure)에 군집화 알고리즘(Clustering algorithm)을 활용하여 유사한 메모리 내의 데이터들을 구분하고, 2) 각 군집의 중심(Centroid)에 하드 주의 기제로 질문 문장과 유사한 K 개의 군집을 특정 지은 후, 3) 해당 군집 내에서 소프트 주의 기제를 통해 유사도가 높은 메모리 내의 데이터를 결정하게 된다. 계층적 메모리 네트워크는 향후 방대하게 증가할 메모리에 주의 기제를 적용할 때 발생할 문제점을 예측하고, 그 문제를 다루기 위한 접근 방식을 성능과 연산 효율 개선에 초점을 두었다는 점에서 주목할 만하다.

3.5 메모리 네트워크 확장 연구 정리

Table 4는 메모리 네트워크를 기반으로 한 연구들을 모델에 따라 분류하여 정리한 표이다.

Table 4. Memory Networks and Derived Models

Model	Attention	Learning task	Dataset	Characteristics
Memory Networks[3]	Hard	Strongly supervised	Large-scale[7], Simulated world QA, SimpleQuestions, bAbI	Proposal of basic QA model
End-to-End Memory Networks[12]	Soft	Weakly supervised	bAbI, Penn TreeBank[25], Text8[26], CBT, CNN QA[27], The Movie Dialog Dataset, The Second Dialog State Tracking Challenge[28]	Less supervision, Inference by using multiple hops
Key-Value Memory Networks[19]	Soft	Weakly supervised	WikiMovies, WikiQA[29]	Key-Value memory representation
Dynamic Memory Networks[21]	Soft	Strongly supervised	bAbI, Stanford Sentiment Treebank(SST)[30], Penn TreeBank	Iterative attention for using Episodic memory module, Applying to 3 NLP tasks
Dynamic Memory Networks+[22]	Soft	Weakly supervised	bAbI, DAQUAR[31], MS COCO Dataset	Multi-modality, Changes for appropriate input module
Recurrent Entity Networks[23]	Soft	Weakly supervised	bAbI, CBT	GRU Entity, Dynamic parallel memory
Hierarchical Memory Networks[24]	Hard, Soft	Strongly supervised	SimpleQuestions	Hybrid of soft and hard attention, Approximate MIPS

4. 뉴럴 튜링 머신 확장 연구

4.1 다양한 주소지정 메커니즘

뉴럴 튜링 머신에서 주소지정 메커니즘은 연산의 복잡도를 결정하는 주요 부분으로, 모델의 성능에 큰 영향을 줄 수 있다. 이에 따라 확장 연구들은 주소지정 메커니즘을 수정하거나 더욱 정교한 과정을 추가하는 등 주소지정 메커니즘 개선에 집중되었다.

주소지정 메커니즘의 효율을 증대시키기 위한 방법은 메모리에 접근한 순서와 관련된 연구들이 주를 이루고 있다. 특히 [32, 33]의 연구는 일반적으로 캐시와 페이지 교체 알고리즘에 사용되는 최소 최근 사용 LRU(Least Recently Used) 기법을 활용하고자 하였다. [32]는 Omniglot 데이터셋의 문자 분류 능력에 대해 뉴럴 튜링 머신과 kNN 분류기의 성능을 비교하여 그 실용성을 확인하였다. 이때 주소지정 메커니즘에 최소 최근 사용 접근 기법을 기존의 내용기반 메커니즘에 추가적으로 적용한다. 이를 통해 새로운 정보를 오랫동안 사용되지 않은 메모리 위치에 쓰거나, 연관된 데이터가 있을 경우 그 위치에 메모리를 업데이트하도록 주소지정 메커니즘을 개선하였다.

동적 뉴럴 튜링 머신(Dynamic Neural Turing Machines) [33]은 기존의 뉴럴 튜링 머신과 다른 세 가지 방법의 주소지정 방법을 제시하였다. 1) 기존의 메모리 행렬에 학습에 용이한(Learnable) 주소 벡터(Address vector)를 추가하여, 위치기반 주소지정 메커니즘의 효율성 증가를 기대했다. 2) 내용기반 주소지정과 최소 최근 사용 접근 주소지정[32]을 동적으로 사용하여 주소지정 기법을 최적화하고자 하였다. 3) 선택적으

로 하드 주의 기제의 이산적 주소지정(Discrete addressing)을 사용하여, 성능 향상을 이루고자 했다. 동적 뉴럴 튜링 머신은 bAbI 태스크와 알고리즘적 문제의 해결 능력을 보이며 기존의 뉴럴 튜링 머신보다 성능의 향상을 증명하였다. 하지만 새로운 방법에 대한 제안과 여러 다양한 시도에 비해 모델의 복잡도가 이슈가 될 수 있으며, 제시한 구조에 대해 타당한 이유가 부족하다.

미분가능 뉴럴 컴퓨터(Differentiable Neural Computer) [34]는 뉴럴 튜링 머신의 주소지정 메커니즘에 정교하고 직관적인 절차를 추가하여 확장한 모델이다. 기존 순환 신경망 기반 모델은 특정한 정보를 기억하기 위해서 사용되는 공간이 연산과 구분되지 않고, 변수나 자료구조 형태의 데이터를 활용할 수 없다는 한계점을 가진다. 이에 반해 미분가능 뉴럴 컴퓨터에서는 데이터를 메모리에 저장하고 이를 자료구조 형태로 만들어 알고리즘적 문제 해결에 기여했다.

‘Differentiable Neural Computer’라는 명칭은 모델이 내포하는 특징을 함축적으로 표현하고 있다. ‘Computer’는 현대 컴퓨터의 중앙처리장치(CPU)와 메모리의 관계처럼 연산(Computation)이 일어나는 곳과 별개의 저장(Storage) 관계를, 연산이 일어나는 망(Networks)과 별개로 저장 공간인 외부 메모리의 관계로 표현한다. 이때 사용되는 망이 ‘Neural Networks’이며, 외부 메모리에 저장되는 값과 관련 연산이 모두 연속적이기 때문에 ‘Differentiable’ 하다.

Fig. 10은 미분가능 뉴럴 컴퓨터의 구조이다. 미분가능 뉴럴 컴퓨터는 뉴럴 튜링 머신처럼 제어기, 읽기/쓰기 헤드, 외부 메모리와 주소지정 메커니즘 확장을 위한 추가 요소들로 구성되어 있다.

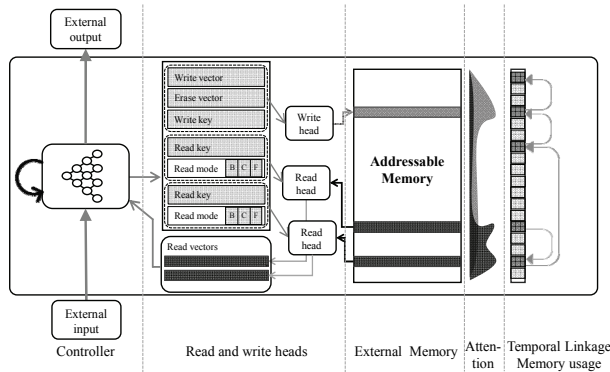


Fig. 10. Differentiable Neural Computer

- 제어기는 변형된 장단기 기억 모델을 적용하여 주소지정 메커니즘에 필요한 가중치를 결정한다. 장단기 기억 모델은 매 시간마다 입력 벡터와 읽기 헤드로부터의 읽기 벡터를 입력 값으로 받고, 그에 따라 출력 벡터와 인터페이스 벡터(Interface vector)를 생성한다. 입력 벡터와 출력 벡터는 매번 입력하고 출력해야 하는 요소는 아니며, 출력 벡터는 메모리 행렬의 영향을 받아 최종적인 결과 값을 만들어낸다. 장단기 기억 모델에 의해 생성된 인터페이스 벡터는 주소지정 할 메모리 위치를 결정하기 위해 사용되며, 인터페이스 벡터를 요소(Element) 단위로 나누어 주소지정에 필요한 많은 요소들을 생성한다.
- 읽기/쓰기 헤드는 메모리의 특정 위치에 읽기/쓰기 연산을 수행한다.
- 외부 메모리는 뉴럴 튜링 머신의 메모리처럼 주소지정 가능한 메모리이다.
- 뉴럴 튜링 머신과 달리, 시간적 메모리 연결(Temporal memory linkage)과 동적 메모리 할당(Dynamic memory allocation)을 위한 메모리 사용량(Memory usage)을 추가 요소로 활용하여 주소지정 메커니즘을 확장한다.

미분가능 뉴럴 컴퓨터의 메모리로부터 읽기/쓰기 연산의 절차는 뉴럴 튜링 머신과 같다. 하지만 읽기/쓰기 연산에 동일한 가중치를 사용하는 뉴럴 튜링 머신과 달리, 미분가능 뉴럴 컴퓨터에서는 각각 다른 가중치를 사용한다. 또한 미분가능 뉴럴 컴퓨터는 내용기반 주소지정과 시간적 메모리 연결을 사용하여 읽기 가중치를 생성해내고, 내용기반 주소지정과 동적 메모리 할당을 사용하여 쓰기 가중치를 생성한다. 생성된 각각의 가중치를 바탕으로 읽기/쓰기 연산에 필요한 읽기 벡터와 쓰기 벡터를 각각 생성한다.

내용기반 주소지정은 뉴럴 튜링 머신의 내용기반 주소지정처럼, 찾고자 하는 키 벡터(Key vector)와 메모리 행렬에서 각각의 벡터와 코사인 유사도를 측정하는 방법이다.

시간적 메모리 연결은 메모리에 쓰기 연산 적용 순서 정보를 활용하기 위한 주소지정 방식이다. 알고리즘적 문제 해결

을 위해서는 메모리 위치 인덱스보다 쓰기 연산의 순서에 따른 메모리 위치 정보가 더 중요한 요인이 된다. 따라서 미분가능 뉴럴 컴퓨터는 시간적 메모리 연결 관계의 행렬을 따로 유지하여, 특정 메모리 위치의 이전에 쓰인 메모리 위치에 대한 가중치(Forward weighting)와 이후에 쓰인 메모리 위치의 가중치(Backward weighting)를 사용한다. 읽기 연산은 1) 내용기반, 2) 이전에 쓰인 메모리 위치에 대한 가중치, 3) 이후에 쓰인 메모리 위치에 대한 가중치, 세 요소의 게이트 연산을 통해 수행된다. 결과 값은 읽기 벡터이며, 생성된 읽기 벡터는 제어기의 입력 값으로 사용된다.

동적 메모리 할당은 사용량 벡터(Usage vector)을 바탕으로, 메모리 할당과 메모리 해제(Free)를 위한 주소지정 방식이다. 사용량 벡터는 메모리 위치의 사용 정도를 나타내는 벡터이다. 따라서 사용량 벡터는 쓰기 연산 시에 값이 증가하며, 해제 게이트(Free gate)에 의해서만 값이 감소할 수 있다. 해제 게이트는 특정 메모리 위치의 해제 정도를 결정하는 요소로, 읽기 연산에 사용된 읽기 가중치를 통해 결정된다. 쓰기 연산은 1) 새로운 메모리 위치로의 할당, 2) 내용기반 주소지정에 의한 위치 결정, 3) 쓰기 연산이 일어나지 않도록 하는 가중치, 세 요소의 게이트 연산으로 결정된다.

미분가능 뉴럴 컴퓨터는 구조체를 형성하여 알고리즘적 문제를 해결했다는 점에서 기여가 크다. 임의생성 그래프(Random-generated graph) 생성을 바탕으로 런던 언더그라운드(Underground)를 형성하여 순회(Traversal), 최단 경로(Shortest-path) 문제를 해결하고, 가계도(Family tree)를 형성하여 학습을 통해 관계 추론(Relation inference)을 수행하는 능력을 보였다. 학습에 사용하는 임의생성 그래프의 형태는 세 자리 숫자로 레이블링(Labeling)되어있는 노드(Node)와 엣지(Edge)로 구성되어 있다. 임의적으로 레이블링 된 노드와 엣지를 생성하여 구조체를 형성하고, 레이블링 된 값을 문제에 따라서 적절하게 입력/출력하여 학습과 테스트가 이루어진다. [34]는 테스트 시에 메모리 위치 정보와 읽기/쓰기 헤드의 작동을 상세하게 시각화함으로써, 미분가능 뉴럴 컴퓨터의 연산 과정을 직관적으로 표현하였다. 뿐만 아니라 bAbI 태스크를 학습하여 메모리 네트워크 기반의 모델에 못지않은 성능 결과를 통해 자연어 처리 문제 해결 능력을 증명하였다. 미분가능 뉴럴 컴퓨터가 모든 문제에서 좋은 성능을 보인 것은 아니지만, 주소지정 가능한 메모리를 바탕으로 구조체를 형성하고 적절하게 사용할 수 있도록 학습했다는 것만으로도 기여가 크다.

미분가능 뉴럴 컴퓨터로 수행한 태스크 중 bAbI 태스크를 제외한 모든 태스크는 커리큘럼 학습(Curriculum learning) [35]이 적용되었다. 인간 혹은 동물이 어떤 것을 배울 때 가장 기본적이고 쉬운 개념에서부터 점차 확장하면서 심화된 것을 배우는 것처럼, 쉬운 학습으로부터 시작해서 점차 어려운 학습을 적용하는 것을 커리큘럼 학습이라 한다. 예를 들

어, 순회 태스크에서는 그래프의 노드 개수 범위, 노드 당 엣지의 개수 범위, 경로의 길이 범위를 학습이 진행됨에 따라 점차 늘려 태스크의 복잡성을 증가시켰다. 이러한 커리큘럼 학습의 적용은 일관적인 학습과정보다 개선된 학습 능력을 보였다.

4.2 강화 학습을 적용한 모험 훈련

주소지정 메커니즘 개선 이외에도, 강화학습을 통해 뉴럴 튜링 머신을 개선하려는 연구가 수행되고 있다. 특히 뉴럴 튜링 머신의 아이디어를 착안한 모델에 강화 학습을 적용하는 연구나 뉴럴 튜링 머신 기반의 모델에 강화 학습을 적용하여 문제를 해결한 연구가 제안되었다.

뉴럴 튜링 머신의 아이디어를 착안하여 외부 인터페이스(External interface)를 추가하고, 외부 인터페이스와의 상호작용에 강화 학습을 적용하고자 한 강화 학습 뉴럴 튜링 머신(Reinforcement Learning Neural Turing Machines)[36]에 대한 연구가 수행되었다. 단순 신경망 모델보다 더 표현적인 능력을 갖기 위한 방법 중 하나는 학습을 통해 외부 인터페이스와 상호작용을 가능하게 하는 것이다. 외부 인터페이스는 메모리, 데이터베이스, 검색 엔진 등 다양한 요소가 가능하다. 뉴럴 튜링 머신은 미분가능한 메모리를 사용하기 때문에 오류 역전파 알고리즘을 사용한 지도 학습이 가능했다. 반면 강화 학습 뉴럴 튜링 머신은 이산적 메모리(Discrete memory)를 사용하여 이를 강화 학습 적용을 통해 학습시키고자 했다. 즉, 강화 학습 뉴럴 튜링 머신은 기존의 뉴럴 튜링 머신을 변형한 것이 아니라 뉴럴 튜링 머신의 아이디어를 모티브로 하고 다른 형태의 메모리를 적용하고자 했다. 하지만 메모리 접근 패턴을 강화 학습으로 학습시키기에는 한계가 있었고, 이론상으로 튜링-완전하지만 실제로 튜링-완전을 측정할 수 없었다는 한계점을 갖는다.

앞서 4.1절에서 설명한 동적 뉴럴 튜링 머신[33]은 강화 학습 뉴럴 튜링 머신[36]에서 이산적 메모리를 사용하는 아이디어를 착안하였다. 따라서 이산적 메모리를 효과적으로 학습하기 위해 강화 학습을 적용했다. 이때 REIN-FORCE[37]를 사용하여 이산적인 형태의 메모리를 강화 학습했다는 것이 특징이다.

또한 미분가능 뉴럴 컴퓨터[34]는 구조체를 형성하여 알고리즘적 문제를 해결하는데 기여했고, 더불어 강화 학습을 적용하여 3X3 크기의 블록 퍼즐 태스크(Block puzzle task)도 해결할 수 있음을 보였다. 블록 퍼즐 태스크는 블록들의 위치 조건을 설명하는 제약 조건(Constraints)들을 모은 하나의 목표(Goal)를 만족시키기 위해 블록을 이동하는 태스크이다. 학습 시에 여러 목표들이 주어지고 그 중 하나의 목표를 제시했을 때, 목표를 만족시키기 위해 블록들을 이동시킨다. 블록은 각 열(Column)에 스택 형식으로 쌓아 올리고 가장 위의 블록만이 다른 열로 이동할 수 있다. [34]는 미분가능 뉴럴

컴퓨터에 강화학습을 적용하여 블록 퍼즐 태스크를 수행하기 위한 실험을 진행했다.

블록 퍼즐 태스크를 수행하기 위한 강화 학습은 두 개의 미분가능 컴퓨터 망으로 수행된다. 1) 첫 번째는 정책 망(Policy networks)이다. 정책 망은 태스크의 환경 상태를 주시하고 제약 조건을 고려해서 블록 이동의 행동(Action)을 확률 분포를 통해 선택한다. 먼저 각 목표와 목표에 해당하는 제약조건들을 정책 망에 입력하게 된다. 입력 시기에 정책 망은 출력과 블록 이동이 일어나지 않는다. 제시되었던 목표들 중 한 목표가 마지막에 제시되면 이진 플래그로 지정된 시작 신호(goal)가 해당 목표에 대해 설정 되고, 정책 망은 이러한 시작 신호가 설정되었는지 주시한다. 특정 목표의 시작 신호가 도착하면, 정책 망은 이 목표를 만족시키기 위해 행동의 확률 분포에 따라 출력을 결정짓게 된다. 블록 퍼즐 태스크 또한 커리큘럼 학습이 적용되었는데, 학습 초기에 정책 망이 결정한 행동의 개수를 늘려서 목표를 만족시키기 쉽도록 학습이 진행되었다. 입력에 필요한 각 블록의 구분, 제약조건 구분, 목표의 구분은 원-핫 인코딩(One-hot encoding)으로 이루어진 벡터가 사용되었다. 2) 두 번째는 가치 망(Value networks)이다. 가치 망은 정책 망과 현재 상태가 주어졌을 때 예측되는 미래 보상의 합계를 추정한다. 다시 말해, 가치 망은 정책 망이 결정한 행동에 따라서 적절한 보상을 한다. 이는 현재 목표의 제약조건 중 만족된 제약조건의 수에 대한 값과 블록이 없는 열에서 블록을 선택하는 것처럼 타당하지 않은 동작을 발생시킨 비용을 뺀 값이다. 기존에는 블록 퍼즐 태스크를 지도 학습 기반으로 해결하고자 했다면, [34]는 두 미분가능 뉴럴 컴퓨터 망을 활용한 강화 학습을 적용하여 뉴럴 튜링 머신 기반 모델의 다양한 학습 적용가능성을 입증하였다.

4.3 희소 접근 방식을 통한 복잡도 개선

뉴럴 튜링 머신은 주소지정에 따른 엄청난 연산량으로 인해 학습이 어려운 단점이 있다. 특히 선형적 계산 부하로 인해 큰 메모리가 필요한 태스크에 적용하기 어렵다. 이러한 문제점을 해결하기 위해, 새로운 뉴럴 메모리 접근 방식을 갖는 희소 접근 메모리(Sparse Access Memory)[38]를 적용하여 연산량을 줄이기 위한 연구가 있다. 뉴럴 튜링 머신에서 모든 메모리 벡터와 연산하는 것과 달리, 희소 접근 메모리는 상수 개의 메모리 벡터에만 접근하여 선형적 계산 부하를 없애는 것이 핵심이다.

희소 접근 메모리의 읽기는 근사적 최근접 이웃 탐색(Approximate nearest neighbor search) 기법 중 임의의 k-d 트리(Randomized k-d tree)와 지역-민감 해시(Locality-Sensitive hash)를 사용하여 k개의 메모리 벡터에만 접근한다. 이를 통해 준선형적(Sublinear) 연산만으로 기존과 비슷한 읽기 벡터를 생성할 수 있다. 또한 메모리 쓰기는 상대적 시간 접근(Relative temporal access) 순서를 저장하고, k+1개

Table 5. Neural Turing Machines and Derived Models

Model	Addressing mechanism	Tasks	Characteristics
Neural Turing Machines[4]	Content-based (Cosine similarity), Location-based (Interpolation, Convolutional shift, Sharpening)	Copy, Repeat copy, Associative recall, Dynamic N-Grams, Priority sort	Neural Networks by coupling External memory, Addressable memory, Selective soft attention
Neural Turing Machines[32]	Content-based (Cosine similarity, Least Recently Used Access(LRUA))	Omniglot, Gaussian Process	Applying to one-shot learning, Reasonable mechanism (LRUA)
Dynamic Neural Turing Machines[33]	Switching between Cosine similarity, LRUA and Discrete	bAbI, Sequential pMNIST, Stanford Natural Language Inference(SNLI)[39], Associative recall, Copy	Trainable memory addressing scheme, Continuous + Differentiable or Discrete + Non-differentiable
Differentiable Neural Computer[34]	Cosine similarity, Dynamic memory allocation, Temporal memory linkage	Random-generated graph (Traversal, Short-path, Family tree(relation inference)), bAbI, Block puzzle (Mini-SHRDLU)	Elaborate and intuitive addressing mechanism, Proposal of next-generation computer model, Troubleshot algorithmic tasks through data structure, Applying to Natural Language Processing, Reinforcement learning
Reinforcement Neural Turing Machines[36]	-	Copy, Repeat copy, DuplicatedInput, Reverse, Forward reverse	Discrete + Non-differentiable, Usability of external interface, Reinforcement learning

의 메모리 벡터에만 접근함으로써 준선형적 연산이 수행되었다. 그 결과, 읽기/쓰기 연산에 상수 시간/공간을 사용하여 기존의 뉴럴 튜링 머신과 유사한 성능을 보였다.

희소 접근 메모리는 종단간 메모리 스키마를 유지한 채 기존 뉴럴 튜링 머신 모델보다 약 3000배 적은 메모리를 읽음으로써 약 1000배 빠른 연산에도 불구하고 비슷한 성능을 내도록 하는 결과를 보였다. 추가적으로 미분가능 뉴럴 컴퓨터에 희소 접근 메모리 기법을 더한 희소 미분가능 뉴럴 컴퓨터(Sparse Differentiable Neural Computer)를 구조화하여 테스트하기도 하였다. 이처럼 희소 접근 방식을 활용하여 기존 모델의 복잡도를 개선하기 위한 시도가 있었고, 연구가 단순히 모델의 정확도에만 치중하는 것이 아니라 학습의 복잡도 개선에 초점을 두었다는 점이 주목할 만하다.

4.4 뉴럴 튜링 머신 확장 연구 정리

Table 5는 뉴럴 튜링 머신을 기반으로 한 연구들을 모델에 따라 분류하여 정리한 표이다.

5. 토 의

메모리 네트워크 기반의 모델은 질의-응답 문제에서 뛰어난 성과를 보였다. 메모리 네트워크는 문맥을 메모리 요소에 저장하고, 신경망의 연산을 통해 정답을 추론할 수 있다. 순환 신경망 구조와 메모리 요소를 결합한 모델의 제안을 시작으로 하여, 종단간 학습이 가능하도록 효율성을 높인 종단간 메모리

네트워크가 제안되었다. 그리고 메모리 네트워크 모델을 대화 시스템에 활용하거나, 질의-응답 문제의 범위를 다방면으로 확장했다. 또한 메모리 모듈을 확장하여 자연어처리 하위 문제를 질의-응답 문제에 적용하거나 병렬적인 메모리 업데이트 구조를 제안하였으며, 이외에도 모델의 구조를 변형하여 복잡도를 개선하려는 시도가 있었다. 이러한 연구들의 성능 지표 평가를 위해 다양한 데이터셋도 개발되었다.

하지만 현재 메모리 네트워크는 한정적인 문법의 사용과 제한된 도메인에서만 해결 능력을 보인다. 복잡한 구조로 이루어진 문장이 대용량으로 주어지는 복합적인 도메인에서는 질의-응답 문제 해결에 대한 연구방향이 아직도 무궁무진하다. 그에 따라 성능을 평가하기 위한 데이터셋도 제안되어야 한다. 또한 현재의 연구 방향이 일반적으로 모델의 정확도에 집중되어 있다. 오히려 효율적인 학습 방법이나 메모리 사용 방식의 변경을 통해 모델의 연산 복잡도를 줄이는 연구가 필요할 것으로 판단된다.

뉴럴 튜링 머신은 주소지정 가능한 메모리의 적절한 위치를 선택하여 읽기와 쓰기 연산을 수행하고, 이를 통해 알고리즘적 문제의 해결 능력을 보였다. 이때 주소지정 메커니즘이 뉴럴 튜링 머신의 가장 핵심적인 부분이지만, 주소지정으로 인해 학습이 어렵고 많은 시간이 소요되는 문제점이 있었다. 따라서 주소지정 메커니즘을 단순화 하거나 직관적이고 합리적인 연산을 추가한 연구가 주를 이뤘다. 또한 뉴럴 튜링 머신 기반의 모델에 강화 학습을 적용하거나 모델의 복잡도를 개선하기 위한 방법론을 제시하였다.

뉴럴 튜링 머신 기반의 모델은 주소지정 메커니즘을 활용하여 문제 해결 방식을 학습시켰다는 것만으로도 기여를 보였다고 할 수 있다. 하지만 성능적인 측면에서 아직 부족한 결과를 보이기 때문에 심화 문제의 학습을 위해 주소지정 메커니즘의 효율적인 대안을 고려하거나, 다른 외부 요소를 추가적으로 활용하는 흥미로운 연구를 기대해 볼 수 있다. 또한 현재까지 주소지정 메커니즘에 연구가 집중되어 있는 것과 반대로, 제어기에 적용할 순환 신경망 구조를 수정하거나, 읽기/쓰기 헤드의 개수를 조정하여 좀 더 병렬적이고 분산적인 구조의 모델 연구에 대해서 고려해 볼 수 있다.

스택-중대 신경망은 메모리를 스택, 큐, 텍, 그리고 리스트와 같은 자료구조 형태로 구조화한 메모리-중대 신경망의 한 분류이다. 스택-중대 신경망은 앞선 메모리 네트워크와 뉴럴 튜링 머신 기반의 모델보다 단순한 구조를 통해 입력 순차 데이터 내의 패턴을 학습하는 단순 알고리즘적 문제를 효율적으로 해결하였다.

스택-중대 신경망은 뉴럴 튜링 머신 기반 모델의 하위 모델이라고 판단할 수 있다. 구조화된 메모리와의 상호작용을 통해 알고리즘적 문제를 해결하지만, 이를 일반화하여 특정한 자료 구조가 아닌 메모리 요소로의 주소 지정 메커니즘을 통해 심화 알고리즘 문제를 해결한 것이 뉴럴 튜링 머신이기 때문이다. 따라서 최근에는 스택-중대 신경망을 확장하는 시도보다 뉴럴 튜링 머신의 주소 지정 메커니즘을 활용하여 확장하는 연구들이 이루어지고 있다.

스택-중대 신경망은 메모리의 형태가 구조화 되어있기 때문에 특정 태스크에만 부합하는 성능을 보이는 한계점이 있다. 이러한 문제점으로부터, 연구에서 제시된 자료구조 이외에도 트리 구조이나 그래프 구조 등 다른 자료구조로 구조화한 메모리의 활용을 고려해볼 수 있다.

메모리 네트워크와 뉴럴 튜링 머신 모델의 질의-응답 문제 해결 능력은 응용프로그램으로 활용 가능하다는 장점이 있다. 최근 상용화되고 있는 인공지능 스피커는 사용자에게 요청이 들어왔을 때 이에 따른 최적의 답변을 도출하는 구조이다. 이는 질의-응답 문제의 한 형태라고 볼 수 있으며, 질의-응답 문제에 최적화된 구조로서 메모리-중대 신경망의 활용 방안을 고려해볼 수 있다. 이에 추가적으로 불완전한 문장, 학습하지 못한 단어, 시간에 흐름에 따른 데이터 축적 등 사용자에게 의해 발생할 상황을 고려하여, 제한적인 질의-응답 시스템으로부터 벗어나기 위한 연구가 필요하다. 뉴럴 튜링 머신은 주소지정 가능한 메모리를 사용하는 점이 다른 문제의 학습으로 확장할 수 있는 주 요소이다. 뉴럴 튜링 머신 기반의 모델은 주소지정 가능한 메모리를 활용하여, 알고리즘적 문제뿐만 아니라 질의-응답 문제도 학습 능력을 보이면서 자연어 처리에 적용 가능성을 증명했다. 따라서 알고리즘적 문제와 자연어 처리 외에 이미지, 음성 인식, 기계 번역 등 순환 신경망 모델을 활용했던 분야로의 세부적인 응용을 기대할 수 있다. 이를

통해 많은 분야에서 널리 사용되는 순환 신경망 모델을 메모리-중대 신경망이 대체 가능할 것으로 보여진다.

6. 결 론

순환 신경망과 장단기 기억 모델의 장기 기억 능력 결여 문제를 해결하기 위해, 메모리 요소를 활용하여 신경망과 상호작용할 수 있도록 학습하는 메모리-중대 신경망 모델 연구가 최근 딥 러닝 분야의 화두로 떠올랐다. 대표적인 메모리-중대 신경망 모델은 질의-응답 문제에 최적화된 메모리 네트워크, 주소지정 가능한 메모리를 학습하여 알고리즘적 문제를 해결하는 뉴럴 튜링 머신, 그리고 구조화된 메모리를 통해 비교적 단순한 구조의 모델로 패턴 학습 알고리즘 문제를 효율적으로 해결하는 스택-중대 신경망, 세 가지로 나눌 수 있다. 이러한 메모리-중대 신경망 모델을 기반으로 하여 여러 파생 연구들이 활발하게 이루어지고 있다.

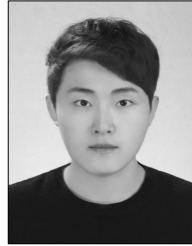
하지만 아직까지는 메모리를 활용한 신경망 모델 연구가 시발점에 위치했다고 할 수 있다. 현재까지 제안된 모델이 인간과 견줄 수 있는 능력을 보이기에선 정확도 측면과 연산량의 효율성에서 한계가 있다. 추후에는 메모리-중대 신경망을 기반으로 모델의 학습 능력 및 효율성 측면의 확장 연구를 기대할 수 있다. 또한 순환 신경망이 여러 분야에서 널리 활용되었듯이, 메모리-중대 신경망이 타 분야에 접목되어 사용되는 일이 필연적이라고 판단된다. 이에 메모리-중대 신경망의 활용 방안과 확장 연구를 탐색하는 것이 유망한 연구 분야로 전망된다.

References

- [1] S. Y. Hea and E. G. Kim, "Design and implementation of the differential contents organization system based on each learner's level," *The KIPS Transactions: Part A*, Vol.18, No.6, pp.19-31, 2011.
- [2] S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach," 3th ed., New York: Prentice Hall, 2009.
- [3] J. L. Hennessy and D. A. Patterson, "Instruction-level parallelism and its exploitation," in *Computer Architecture: A Quantitative Approach*, 4th ed., San Francisco, CA: Morgan Kaufmann Pub., ch. 2, pp.66-153, 2007.
- [4] D. B. Lenat, "Programming artificial intelligence," in *Understanding Artificial Intelligence*, Scientific American, Ed., New York: Warner Books Inc., pp.23-29, 2002.
- [5] A. Stoffel, D. Spretke, H. Kinnemann, and D. A. Keim, "Enhancing document structure analysis using visual analytics," in *Proceedings of the ACM Symposium on Applied Computing*, Sierre, pp.8-12, 2010.
- [6] J. Y. Seo, "Text driven construction of discourse structures for understanding descriptive texts," Ph.D. Dissertation, University of Texas at Austin, TX, USA, 1990.

- [7] Thomas Claburn, Google Chrome 18 Brings Faster Graphics [Internet], <http://www.techweb.com/news/232800057/google-chrome-18-brings-faster-graphics.html>.
- [8] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S.Khudanpur, "Recurrent neural network based language model," In *Interspeech*, pp.1045-1048, 2010.
- [9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol.9, No.8, pp.1735-1780, 1997.
- [10] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.
- [11] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.
- [12] A. Joulin and T. Mikolov, "Inferring algorithmic patterns with stack-augmented recurrent nets," In *Advances in Neural Information Processing Systems*, 2015.
- [13] E. Grefenstette, K. M. Hermann, M. Suleyman, and P. Blunsom, "Learning to Transduce with Unbounded Memory," In *Advances in Neural Information Processing Systems*, 2015.
- [14] A. Fader, L. Zettlemoyer, and O. Etzioni, "Paraphrase-driven learning for open question answering," In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp.1608-1618, 2013.
- [15] A. Bordes, N. Usunier, R. Collobert, and J. Weston, "Towards understanding situated natural language," In *Proceedings of Machine Learning Research*, pp.65-72, 2010.
- [16] S. Das, C. L. Giles, and G. Sun, "Learning context-free grammars: Capabilities and limitations of a recurrent neural network with an external stack memory," In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 1992.
- [17] G. Z. Sun, C. L. Giles, H. H. Chen, and Y. C. Lee, "The neural network pushdown automaton: Model, stack and learning simulations," *UNIVERSITY OF MARYLAND TR NOs. UMIACS-TR-93-77 & CS-TR-3118*, 1993.
- [18] K. Cho, B. Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp.1724-1734, 2014.
- [19] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," In *Advances in Neural Information Processing Systems*, pp.2431-2439, 2015.
- [20] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. H. Miller, A. Szlam, and J. Weston, "Evaluating prerequisite qualities for learning end-to-end dialog systems," *arXiv preprint arXiv:1511.06931*, 2015.
- [21] J. Weston, "Dialog-based language learning," In *Advances in Neural Information Processing Systems*, 2016.
- [22] A. Bordes, Y. Boureau, and J. Weston, "Learning End-To-End Goal-Oriented Dialog," *arXiv preprint arXiv:1605.07683*, 2016.
- [23] A. Bordes, N. Usunier, S. Chopra, and J. Weston, "Large-scale simple question answering with memory networks," *arXiv preprint arXiv:1506.02075*, 2015.
- [24] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. Merriënboer, A. Joulin, and T. Mikolov, In *The 4th International Conference on Learning Representations*, 2015.
- [25] F. Hill, A. Bordes, S. Chopra, and J. Weston, "The goldilocks principle: Reading children's books with explicit memory representations," In *The 4th International Conference on Learning Representations*, 2015.
- [26] A. H. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, and J. Weston, "Key-Value Memory Networks for Directly Reading Documents," *arXiv preprint arXiv:1606.03126*, 2016.
- [27] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on Freebase from question-answer pairs," In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.
- [28] A. Kumar, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask Me Anything: Dynamic memory networks for Natural Language Processing," In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.
- [29] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.
- [30] M. Henaff, J. Weston, A. Szlam, A. Bordes, and Y. LeCun, "Tracking the World State with Recurrent Entity Networks," In *The 5th International Conference on Learning Representations*, 2016.
- [31] S. Chandar, S. Ahn, H. Larochelle, P. Vincent, G. Tesauro, and Y. Bengio, "Hierarchical memory networks," *arXiv preprint arXiv:1605.07427*, 2016.
- [32] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The Penn Treebank," *Computational Linguistics*, Vol.19, No.2, pp.313-330, Jun. 1993.
- [33] T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, and M. Ranzato, "Learning longer memory in recurrent neural networks," In *The 4th International Conference on Learning Representations*, 2015.
- [34] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching Machines to Read and Comprehend," In *Advances in Neural Information Processing Systems*, 2015.

- [35] M. Henderson, B. Thomson, and J. Williams, "The second dialog state tracking challenge," In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp.263-272, 2014.
- [36] Y. Yang, W. Yih, and C. Meek, "Wikiqa: A challenge dataset for open-domain question answering," In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp.2013-2018, 2015.
- [37] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp.1631-1642, 2013.
- [38] M. Malinowski and M. Fritz, "A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input," In *Advances in Neural Information Processing Systems*, 2014.
- [39] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.
- [40] C. Gulcehre, S. Chandar, K. Cho, and Y. Bengio, "Dynamic Neural Turing Machine with Continuous and Discrete Addressing Schemes," *arXiv preprint arXiv:1607.00036*, 2017.
- [41] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwinska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, and D. Hassabis, "Hybrid computing using a neural network with dynamic external memory," *Nature*, Vol.538, pp.471-476, 2016.
- [42] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp.41-48, 2009.
- [43] W. Zaremba, and I. Sutskever, "Reinforcement learning neural turing machines," *arXiv preprint arXiv:1505.00521*, 2015.
- [44] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning," *Machine Learning*, Vol.8, No.3-4, pp.229-256, 1992.
- [45] J. W. Rae, J. J. Hunt, T. Harley, I. Danihelka, A. Senior, G. Wayne, A. Graves, and T. P. Lillicrap, "Scaling Memory-Augmented Neural Networks with Sparse Reads and Writes," In *Advances in Neural Information Processing Systems*, 2016.
- [46] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *arXiv preprint arXiv:1508.05326*, 2015.



이 지 환

<https://orcid.org/0000-0002-1274-8185>
 e-mail : ji_hwan43@yonsei.ac.kr
 2017년 연세대학교 컴퓨터과학과(학사)
 2017년~현 재 연세대학교 컴퓨터과학과 석사과정
 관심분야: 인메모리 데이터베이스 시스템, 기계 학습



박 진 옥

<https://orcid.org/0000-0003-0424-8225>
 e-mail : parkju536@yonsei.ac.kr
 2016년 서울시립대학교 통계학과(학사)
 2017년~현 재 연세대학교 컴퓨터과학과 석사과정
 관심분야: 빅데이터 마이닝, 기계 학습



김 재 형

<https://orcid.org/0000-0003-2518-0012>
 e-mail : jaehyungkim@yonsei.ac.kr
 2007년 한양대학교 컴퓨터공학과(학사)
 2009년 한양대학교 컴퓨터공학과(석사)
 2013년~현 재 연세대학교 컴퓨터과학과 박사과정
 관심분야: 데이터베이스, 스토리지, 빅데이터



김 재 인

<https://orcid.org/0000-0002-9524-4010>
 e-mail : james5809@yonsei.ac.kr
 2016년~현 재 연세대학교 컴퓨터과학과 학사과정
 관심분야: 자연어 처리, 기계 학습



노 흥 찬

<https://orcid.org/0000-0001-9892-2561>
 e-mail : hongchan.roh@sk.com
 2006년 연세대학교 컴퓨터과학과(학사)
 2008년 연세대학교 컴퓨터과학과(석사)
 2014년 연세대학교 컴퓨터과학과(박사)
 2014년~현 재 SK Telecom ICK 종합원 매니저
 관심분야: 데이터베이스 시스템, 플래시 SSD



박 상 현

<https://orcid.org/0000-0002-5196-6193>

e-mail : sanghyun@yonsei.ac.kr

1989년 서울대학교 컴퓨터공학과(학사)

1991년 서울대학교 컴퓨터공학과
(공학석사)

2001년 UCLA 컴퓨터과학과(공학박사)

2001년~2002년 IBM T. J. Watson Research Center

Post-Doctoral Fellow

2002년~2003년 포항공과대학교 컴퓨터공학과 조교수

2003년~2006년 연세대학교 컴퓨터과학과 조교수

2006년~2011년 연세대학교 컴퓨터과학과 부교수

2011년~현 재 연세대학교 컴퓨터과학과 교수

관심분야: 데이터베이스, 데이터마이닝, 바이오인포매틱스,
빅데이터 마이닝 & 기계 학습