Memory Augmented Neural Networks

Background

- RNN -> LSTM -> Encoder-Decoder -> Attention mechanism on Encoder Decoder model -> NTM -> MANN
- This model is a modification on the NTM architecture.
- It has both a controller (NN Module) and a memory matrix
- NTM model had both Content based addressing and Location based addressing. MANN is purely content based addressing.
- Least Recently Used Access is used in writing to memory.

Points to Note

- Turn neural networks into a differential computer by giving them read and write access to external memory.
- Everything is differentiable. Controller, Memory and memory interface (heads that select portions of memory to read and write)
- MANN has blurry read mechanism and write mechanism. Read and write weight vectors unlike NTM.
- In encoder-decoder with attention, the decoder at time t has only access to all encoder outputs of current episode (time t) and the decoder output of time t-1. But in MANN, the controller has access to many episode data to refer to.
- Least Recently Used Access mechanism for writing. This module emphasizes accurate encoding of relevant (i.e., recent) information, and pure content-based retrieval.
- Idea of the model : Given x(t), inputted to the controller, outputs h(t) and c(t) produced. h(t) used in finding the weights and blurry read and write take place. Memory vector r(t) is obtained using the read vector and memory matrix, which is then used as input to the classifier, such as a softmax output layer to give output. This memory vector r(t) is similar to the weighted sum of encoder and decoder inputs in attention based Encoder-Decoder architecture.

Weight parameters in the model

- w_key ::- k_t = matmul (h_t, w_key)
- w_sigma ::- sigma_t = matmul(h_t, w_sigma)
- w_xh and w_hh ::-
  Gates (preactivations) = matmul(x_t, w_xh) + matmul(h_tm1, w_hh) + b_h
- w_lutm1 - least used weights
- ww_t - write weights = sigma*wr_tm1 + (1-sigma)*wlu_tm1
- M_t - Memory matrix
- K_t = cosine similarity(k_t, M_t)
- wr_t - read weights = softmax(K_t)
- wu_t - usage weights = gamma*wu_tm1 + wr_t + wu_t
- r_t - read memory = matmul(wr_t, M_t)

Data Augmentation

[Data Augmentation Generative Adversarial Networks]

Summary:

- Standard data augmentation produces only limited plausible alternative data. Given there is potential to generate a much broader set of augmentations, we design and train a generative model to do data augmentation.

Imitation Learning - RL

[One-Shot Imitation Learning]

Meta Learning - learning to learn

[Meta-Learning with Memory-Augmented Neural Networks]

Anomaly Detection

Memory Augmented Generative Adversarial Networks for Anomaly Detection

Question - Answering Tasks

Visual Question Answering task (Image Captioning): Given an image and a natural language question about the image, the task is to provide an accurate natural language answer.

Visual Question Answering with Memory-Augmented Networks

Graph Traversal Tasks

[Memory Augmented Graph Neural Networks for Sequential Recommendation]

Language modelling Tasks

Sentence Simplification with Memory-Augmented Neural Networks

Reasoning with Memory Augmented Neural Networks for Language Comprehension

Neural Machine Translation with Key-Value Memory-Augmented Attention

Handwritten Recognition Tasks

[Improving Long Handwritten Text Line Recognitionwith Convolutional Multi-way Associative Memory]()

TARDIS : https://arxiv.org/pdf/1701.08718.pdf (better MANN model)

Refer

Concept Learning through Deep ReinforcementLearning with Memory-Augmented Neural Networks

scaling-memory-augmented-neural-networks-with-sparse-reads-and-writes

NTM