RESEARCH ARTICLE OPEN ACCESS

Comparison of the performance of Memory Augmented Neural Network Model with Long-Short Term Memory Model

Prof. A.M Chandrashekar[1], Adarsh L[2], Bhavana S[3], Varsha G[4]
[1,2,3,4]Department of Computer Science and Engineering, Sri Jayachamarajendra College of Engineering, Mysore

Abstract:

A lot of breakthroughs have happened in deep neural networks over a period of time. Traditional gradient-based networks require a lot of data to train and learn. When new data is encountered, the models must inefficiently relearn their parameters to effectively learn the new information without any major interference. Neural Turing Machines (NTMs) which have augmented memory capacities, offer the ability to quickly learn and retrieve new information, and hence can potentially remove the disadvantages of conventional models. Here, we demonstrate the ability of a memory-augmented neural network to rapidly assimilate new data, and use this data to make accurate predictions after only a few samples(One-shot). We also introduce a new method for accessing an external memory that focuses on memory content(content-based addressing), unlike previous methods that additionally use memory location based focusing mechanisms. We have also compared accuracy and learning loss of the MANN (Memory Augmented Neural Networks) with LSTM (Long-Short Term Memory) to show which model is a better choice.

Keywords **— Neural Turing Machine, Memory Augmented Neural Networks, Long-Short Term Memory.**

1. INTRODUCTION

One of the efficient object categorization method is One shot learning which comes under computer vision. Traditional approaches of machine learning requires large number of training samples and datasets to achieve acceptable accuracy. One shot learning plays a major role by learning information from one or few samples.

2.BACKGROUND

Majority of classification approaches come with a lot of disadvantages. Challenges of One shot learning approach is listed below:

Challenge of Representation: Modelling of data objects should be done prior to categorization.

Challenge of Learning: Is selection of a method used to acquire that model, so that it can be used for efficient learning.

Challenge of Recognition: Detecting a new image based on previous knowledge acquired during object categorization. Challenge here is detecting in presence of occlusion, viewpoint, and lighting changes

One-shot learning emphasizes on knowledge transfer, which makes use of prior knowledge of learnt categories and allows for learning on minimal training examples. Thus it differs from single object recognition and standard category recognition algorithms.

1) Knowledge transfer by model parameters: One shot learning algorithms use model parameters. This is one set of one-shot learning algorithm which achieves knowledge transfer based on the similarity between previously and newly learned classes. Similarity is achieved by learning several training examples and then the new object classes are learned using transformations of model parameters from the previously learned classes.

2) Knowledge transfer by sharing features: This is the second kind of algorithm which use knowledge transfer by sharing. This is exchanging information on various features of objects with several classes. Bart and Ullman [3] have proposed an algorithm which extracts diagnostic information in patches from alreadylearnt classes by maximizing the patches; this approach applies these features to the learning of a new class. Consider an example of cow class which may be learned in one shot from previous knowledge of parrot and camel classes, since cow objects may contain similar distinguishing patches.

3) Knowledge transfer by contextual information: This method focuses on a broader scale i.e., by appealing on a global knowledge of the scene in which object is projected. D.Hoiem et al. has proposed an algorithm [4] which makes use of contextual information in the form of camera height and scene geometry to prune object detection. Two advantages of these kind of algorithms are: First, learning the object classes which look relatively dissimilar; and second, performing well in situations where an image has not been hand-cropped and carefully aligned, but rather which naturally occur.

3. NEURAL TURING MACHINE(NTM)

An NTM is a neural network controller coupled to external memory resources, with which it interacts. The memory inter-actions are differentiable end to end. The controller of NTM can be a feed - forward network or LSTMs. The controller interacts with an external memory module using a number of read and write heads.

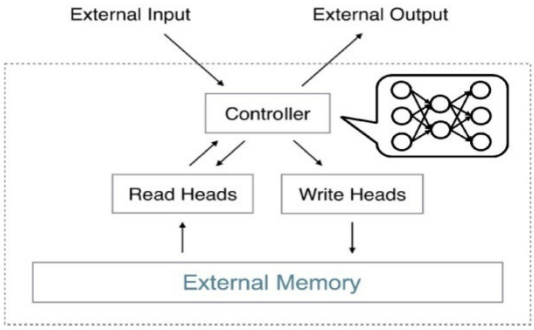

Fig. 1. Neural Turing Machine

Read and write heads help to retrieve information from memory or place new information into memory, respectively. Memory encoding and retrieval in NTMs is rapid. This feature of NTM makes it a perfect candidate for meta-learning [1] and low-shot [1] prediction, since it can be used for both long-term storage which is possible with slow updates of network weights, and short-term storage with its external memory module. The goal here is to modify an NTM model
to excel at one-shot learning: Restricting the controllers ability to write to memory using location based addressing, so that the controller will learn more quickly. Normally, different algorithms use location-based addressing. But, in one shot learning, we make use of content based addressing mechanism. This is because for a given input, there are only two actions the controller might need to do and both depend on content-addressing. One action is that the input is very similar to a previously seen input; in this case, we might want to update whatever we wrote to memory. The other action is that the input is not similar to a previously seen input; in this case, we do not want to overwrite recent information, so we will instead write to the least used memory location.

4. APPLICATIONS

Computer vision is working far better than just two years ago, and this is enabling numerous exciting applications ranging from

1) The design of an autonomous vehicle tool-chain, which captures formal descriptions of driving scenarios in order to develop a safety case for an autonomous vehicle (AV). To achieve this, the robot needs to process complex visual information, and the robot needs to figure out how to act based on what it observes. In order to process visual information, we can benefit from the success in computer vision.

2) A software application which is used in organizations and industries to identify people based on facial characteristics [12] captured in a photo or a video. This is done by comparing distinct facial features from the image or video frame and a database maintained by the organization. There are two variants in this approach: Face verification and Face recognition.

Face verification: Verify whether or not input image is of the claimed person (1:1 problem)

Face recognition: Recognize a person out of k datasets (1:k problem). Former is hard [13] since it has to solve one shot learning problem (Organization normally has only one picture of employees)

3) Interpretation of text and images in medical reports can be achieved using image processing algorithms. They collect signals from various inputs and help patients by suggesting correct diagnosis [16] for particular disease.

4) This tool helps in conversion of handwritten text, typed sentences, or printed symbols into machine-encoded text. Recognizing text plays a major role in many fields including: Postal address envelops, banking transactions, insurance, etc. We will be demonstrating one of the main applications of One shot Learning in this project, i.e., Character recognition.

5. EXISTING SOLUTION METHODS

Traditional machine learning algorithms like gradient-based techniques [7] require a lot of data to learn, often through extensive iterative training. When new data is presented, the learning process is repeated on the new data for classification purpose. This is inefficient and performance drops drastically. In scenarios where only few samples are available and presented one by one for training, gradient-based solution (re-learn the parameters from the data available at the moment) [4] is prone to poor learning and gives catastrophic errors.
In such cases where training data is sparse or not enough to produce expected output, traditional gradient-based solution is to completely re-learn the parameters from the data available at the moment. This is prone to poor learning, and become unreliable for real life scenarios. These hazards are the reason for choosing non-parametric methods for better results. Meta

Learning [1] can be one good solution for such problems. Meta-learning refers to a scenario in which an agent learns at two levels, each associated with different time scales. One, which occurs within a task called Rapid Learning [8]. It could be learning to accurately classify within a particular dataset. This is a kind of learning which acquires knowledge about the class gradually across tasks and the other, which captures the way in which task structure varies across target domains. Given its two-tiered organization, this form of meta-learning is often described as learning to learn. Thus in addition to learning how to solve a particular task, meta learning helps in learning the way task itself is designed.

It is observed that neural networks with augmented memory capacities [10] could be used for meta-learning tasks. These networks learn about the data through weight updates, but also change their output by rapidly storing representations of modules in external memory attached to them.

6. PROPOSED SOLUTION METHODS

1) Memory-Augmented Model: NTM or Neural Turing Machine consists of two main components: a)A controller b)Memory bank (storage). The controller [9] interacts with the external resources with the help of input and output vectors. It can also interact with the memory matrix with the help of read and write heads to perform selective read and write operations. This architecture has components which are easy to differentiate and can help model training with easier gradient descent techniques [14]. This can be achieved by using blurry read and write operations. We have to define the degree of blurriness. Degree of blurriness can be defined as a mechanism that controls read and write heads to focus on a smaller portion of memory while ignoring the rest. The portion of memory focused by heads can be determined by using the outputs emitted by heads. These outputs are characterized by weights over the rows in the memory location. Each weight corresponds to the no. of read and write operations performed in that particular memory location.

Data Set: A standard data set that contains several character symbols called Omniglot dataset [5] which has 1600 symbols. We divide these character classes as 1200 training samples,423 test classes. We also try to create new classes by rotating the samples in different angles like 90,180,270 degrees [15]. In order to reduce the processing time, we also scale the images down to 20x20.

Least recently used access: The least recently used vector w is an element generated by usage weight vector by setting the minimum element in the usage weight vector to 1 while setting all the other elements to 0. For instance, let the usage vector be [0.2, 0.7, 0.4], then the least used weight vector would be [1,0,0].

Character Recognition Using Omniglot Dataset: We consider Omniglot dataset for this project which consists of 50 alphabets or character classes. We split it into background set and evaluation set. Background set would contain 30 classes whereas the evaluation set contains 20 classes.

Background set can be mainly used while training the samples to acquire general knowledge about the characters. Evaluation set can be mainly used while testing the samples and to compare the derived result with the standard one.

7. SYSTEM REQUIREMENTS

6.1 Software requirements

1) Operating system: This project is developed in a system running windows 10 operating system. It can run on any other operating system which supports the tools and technologies mentioned in the next section.

2) Python: The implementation is done using python language using anaconda and python IDLE. Python is a high-level, interpreted, interactive and object-oriented scripting language. It makes use of simple English words which helps beginners use the language in a easy and a better way. The syntax used to define the language has fewer rules when compared to other programming languages. It also features dynamic type system which includes suggestions for the programmer to rectify errors immediately and an automatic memory management to free the memory when not used and many other features.

3) Anaconda: Anaconda is a most popular Python Data Science platform which supports variety of tools, modules, packages which are of great use to programmers. It has Spyder and Jupiter as IDEs, Orange as a data mining tool, IPython Console etc. to name a few. As it is a open source distribution, its widely used.

4) Python IDLE: IDLE is Pythons Integrated Development and Learning Environment. IDLE has the following features:

-Platform Independent: works mostly the same on Windows, Unix, and Mac OS X.

-Python shell window (interactive interpreter) which easily distinguishes the input, output and errors by highlighting them with different colors.

-Multi-window text editor which supports multiple undo, Python colorizing, smart indent, auto completion, and other features search within any window, replace within editor windows, and search through multiple files (grep).

-Debugger with persistent breakpoints, stepping and viewing of local and global namespaces.

6.2 Hardware Requirements

Processor 64 bit, 4 core, 2.00 GHZ RAM 8 GB Hard disk 14 GB for installation. No free disk space required for running

and production use.

8. TOOLS AND TECHNOLOGIES USED

7.1TensorFlow

TensorFlow is an open source software library for numerical computation using data flow graphs. Graphs usually have nodes and edges. Edges are the ones connecting different nodes. So, nodes here represent mathematical operations, whereas edges represent multidimensional data arrays also known as tensors which are communicated between the nodes. The name TensorFlow is derived due to the multidimensional data arrays which interact to perform some operation. These multidimensional data arrays are called 'Tensors'.

7.2 HighCharts

Highcharts is a charting library which is purely based on JavaScript which can be used to improve the web applications and helps to picturize the results obtained. It supports a wide variety of charts. For example, line charts, spline charts, area charts, bar charts, pie charts and so on [11].

7.3 Python Flask

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It supports several libraries and modules which can take care of basic programming details like protocols, thread management.

9. SYSTEM DESIGN

The addressing of memories of NTM can be done by both content-based and location-based. Iterative steps which run along a tape along with the jumps across the memory is used in location-based addressing. This method of addressing is much helpful for the tasks which involve sequence-based prediction. When the tasks that emphasize conjunctive coding of information independent of sequence, this type of access is not optimal. Least Recently Used Access (LRUA) module is used for writing to the memory in our model which is one of the newly designed access modules. It writes data to memories either to the least used memory location or the most recently used memory location. It is a pure content-based memory writer that emphasizes accurate recent information encoding and pure content-based retrieval. Rarely-used locations are over-written by new information which preserves recently encoded information or even it is over-written to the last used location, which can be viewed as an updation of the memory with newer and more relevant information [2].

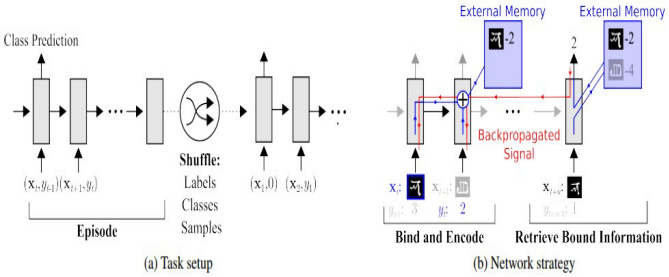

Fig.(3).System Design

(a)Task setup - Omniglot images (or x-values for regression), x_t, are assigned with labels, y_{t-1} , the output predicted class label from the previous step. From one episode to another episode, the classes in the episode, their associated labels, and the specific samples are all shuffled, so that it prevents the network from memorizing the labels and its values.

(b)Network strategy - This strategy would involve the use of an external memory which helps to store bound the sample class label information. When a sample from an already-seen class by the model is presented, then the retrieval from the memory at a later point for successful classification is easier. A sample data at a particular time step x_t should be associated with its appropriate class label y_t, which is essential in the future time step. Whenever a sample from the same class is seen, the model does a proper prediction by retrieving the bound information from the external memory. The shaping of the weights from the previous predicted step is through backpropagated error signals which is the unique form of binding strategy [2].

10. SYSTEM IMPLEMENTATION

One Shot Learning using Memory-Augmented Neural Networks (MANN) is based on Neural Turing Machine architecture. The images from the background set of the Omniglot dataset [5] which consists of images of the characters of 30 different languages are used for the purpose of training the model and the images of the evaluation set of the Omniglot dataset [5] which consists of images of the characters of 20 different languages are used for the purpose of testing. The images are randomly selected from the dataset for each sequence length consisting of 50 samples which is the combination of the samples of 5 different classes. The images are assigned with labels and their corresponding output labels are known for the purpose of training the model. The model predicts the class label which is compared with the exact output label and the weights are adjusted based on the loss value obtained in each iteration of the batch.

1) **Learning of the model:** In one-hot label classification,

the network minimizes the episode loss of the input sequence, given the probabilities output by the network [2].

$$\mathcal{L}(\theta) = -\sum_t \mathbf{y}_t^T \log \mathbf{p}_t, \qquad (1)$$

Where y_t is the target one-hot label at time t. Only one element assumes the value 1 for a given one-hot class-label vector y_t, and five elements assume the value 1, one per five-element chunk, for a string-label vector. The loss for string label classification is given by:

$$\mathcal{L}(\theta) = -\sum_t \sum_c \mathbf{y}_t^T(c) \log \mathbf{p}_t(c). \qquad (2)$$

where (c) indexes a five-element long 'chunk' of vector label of which there are a total of five.

2) **Model Training and Testing:** Every episode consists of 5, 10 or 15 unique classes. Episode lengths is usually kept ten times the number of unique classes (i.e., 50, 100, or 150 respectively), unless the length is explicitly mentioned. Training of the model is done for 100000 episodes. The model is trained effectively by the time training reaches the 100000 episode mark, and weight updates are ceased. Training is one of the important phase where in the problem of under-fitting and over-fitting arises [17]. So, the model has to be trained in such a way that the accuracy should be maximized and at the same time the loss should be minimized. Then the task of testing is done where data are pulled from a disjoint test set (i.e., samples that belong to the classes 1201-1623 in the omniglot dataset).

3) **Input data classification:** Input sequences are flattened, pixel-level representations of images xt and the time-offset labels yt-1. The first N unique classes are sampled from the Omniglot dataset which is given as input, where N is the maximum number of unique classes per episode. N assumes a value of either 5, 10, or 15, which is indicated in the model training and testing phase description. Extracted samples from the Omniglot source set are kept if they are members of the set of N unique classes for that given episode, and otherwise they are discarded. Approximate image data for the episode is kept as 10N samples [2].

11. RESULT ANALYSIS

The output accuracy obtained for 1st, 3rd, 5th and 10th instances of the images and learning loss for different batches of MANN and LSTM model is shown in the tables below.

The accuracy increases greatly from 1st to the 3rd instance, which proves that the MANN model can be trained with high accuracy with one or few number of samples which satisfies the agenda of one-shot learning.

TABLE I
ACCURACY AND LEARNING LOSS DURING TRAINING OF MANN MODEL

1st	3rd	5th	10th	batch	loss
0.2000	0.2125	0.2000	0.2128	0	80.6079
0.2750	0.1392	0.1948	0.2727	1000	80.4200
0.2625	0.6750	0.7595	0.7447	5000	40.9961
0.2000	0.8000	0.9114	0.8205	10000	29.5128
0.2875	0.8750	0.9091	0.9800	25000	16.5823
0.3250	0.9000	0.9610	0.9767	50000	13.3374
0.3875	0.9625	0.9744	0.9787	99900	11.0842

TABLE II
ACCURACY AND LEARNING LOSS DURING TESTING OF MANN MODEL

1st	3rd	5th	10th	loss
0.3150	0.9211	0.9558	0.9668	14.627711

The accuracies of the LSTM at different instances is not greater than that of MANN model accuracies which means that the LSTM model could not be trained with few samples and with small number of batches.

TABLE III
ACCURACY AND LEARNING LOSS DURING TRAINING OF LSTM MODEL

1st	3rd	5th	10th	batch	loss
0.2250	0.1500	0.2532	0.2083	0	80.5391
0.1500	0.2125	0.1772	0.1915	1000	80.5136
0.1875	0.2000	0.1625	0.3043	5000	80.4452
0.2125	0.6750	0.6329	0.8140	10000	52.5396
0.2375	0.7000	0.8481	0.8889	25000	35.4734
0.2000	0.8000	0.8228	0.9091	50000	26.0972
0.3125	0.9125	0.8974	0.8696	99900	18.9330

TABLE IV
ACCURACY AND LEARNING LOSS DURING TESTING OF LSTM MODEL

1st	3rd	5th	10th	loss
0.2534	0.7922	0.8395	0.8673	28.682074

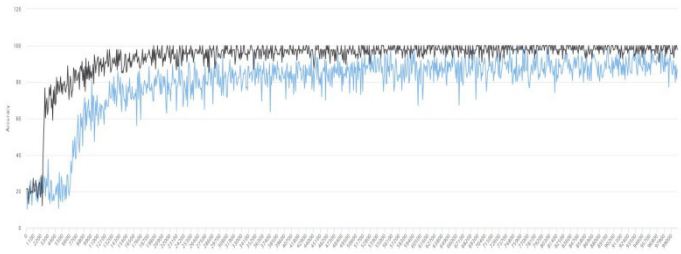

Fig. 3. Comparison of the accuracy during training of MANN and LSTM model

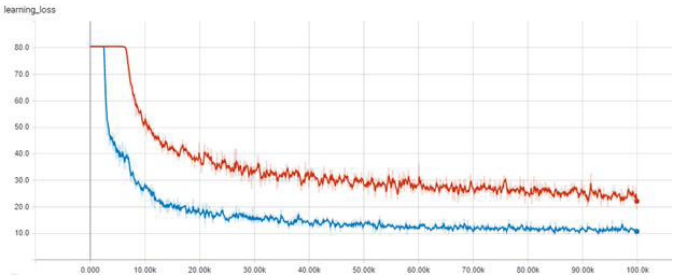

Fig. 4. Comparison of the learning loss during training of MANN and LSTM model

The"Fig. 3" shows the variation of accuracy during the training of MANN (black line) and LSTM model (blue line) for 100000 batches. We observed that the MANN model reached higher accuracy in less number of batches when compared with LSTM model.

In "Fig. 4", the blue line indicates the learning loss of MANN model and the red line indicates the learning loss of the LSTM model. The graph clearly shows that the learning loss of the MANN model decreases faster than the LSTM model as the number of epochs increases, which proves that the model can be trained better than the LSTM model.

CONCLUSION

Memory augmented neural network (MANN) can be used to recognize the characters accurately when only one or few samples are available. MANN model has produced an accuracy of 97% for the testing samples. MANN model has produced a learning loss value of 11.0842 which means that it has been trained effectively. Comparative study of accuracy and learning loss of MANN and LSTM models has provedthat MANN is way better than LSTM.

REFERENCES

[1] Alex Graves, Greg Wayne, Ivo Danihelka, Neural Turing Machine https://arxiv.org/pdf/1410.5401.pdf

[2] Santoro, Adam, et al. One-shot learning with memory-augmented neural network (19 May 2016) https://arxiv.org/pdf/1605.06065.pdf

[3] CVPR 2005 by Bart and Ullman https://www.vision.caltech.edu/~bart/.../2005/BartUllmanCrossGeneralizationInternal.ps

[4] Diagnosing Error in Object Detectors

[5] Omniglot Data Set for one-shot learning https://github.com/brendenlake/omniglot

[6] Graves, Wayne and Danihelka, Explanation of neural turing machine(10 Dec 2014) https://rylanschaeffer.github.io/content/research/neural_turing_machine/main.html

[7] Recurrent Neural Networks Tutorial, Part 3, Back propagation through time and vanishing gradients http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients

[8] Recurrent Neural Network Tutorial, Part 4, Implementing a GRU/LSTM RNN with Python and Theano http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/

[9] The Unreasonable Effectiveness of Recurrent Neural Networks http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[10] Understanding LSTM Networks http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[11] HighCharts tutorial https://www.highcharts.com/docs/getting-started/your-first-chart

[12] A. M. Chandrashekhar, Anjana D, Muktha G, Cyber talking and Cyber bullying: Effects and prevention measures, Imperial Journal of Interdisciplinary Research (IJIR), Volume 2, Issue 2, JAN- 2016. https://www.onlinejournal.in/IJIRV2I3/017.pdf

[13] Rahil Kumar Gupta, A.M.Chandrashekhar, Shivaraj H. P, Role of infor-mation security awareness in success of an organization International Journal of Research(IJR) Volume 2, Issue 6, May 2015.

[14] A. M. Chandrashekhar and K. Raghuveer, Amalgamation of K-means clustering algorithem with standard MLP and SVM based neural net-works to implement network intrusion detection system, Advanced Computing, Networking, and Informatics Volume 2(June 2014), Volume 28 of the series Smart Inovation, Systems and Technologies pp 273-283.

[15] A. M. Chandrashekhar and K. Raghuveer, Fusion of Multiple Data Mining Techniques for Effective Network

Intrusion Detection A Con-temporary Approach, Proceedings of The 5th International Conference on Security of Information and Networks (SIN 2012), 2012, pp 33-37.

[16] A. M Chandrashekhar A M and K. Raghuveer, Diverse and Con-glomerate Modi-operandi for Anomaly Intrusion Detection Systems, International Journal of Computer Application (IJCA) Special Issue on Network Security and Cryptography (NSC), 2011.

[17] A. M. Chandrashekhar and K. Raghuveer, Confederation of FCM Clustering, ANN and SVM Techniques of Data mining to Implement Hybrid NIDS Using Corrected KDD Cup Dataset, IEEE International Conference on Communication and Signal Processing (ICCSP),2014, pp 672-676.