Memory Augmented Neural Network for Extreme Class Imbalance

Project-II (AI67002) report submitted to

Indian Institute of Technology Kharagpur

in fulfillment for the award of the degree of Master of Technology

in

Artificial Intelligence, Machine Learning, and Applications

by

Soumya Ranjan Patra

(18ME3AI30)

Under the supervision of

Professor Jiaul Hoque Paik

Centre of Excellence in Artificial Intelligence Indian Institute of Technology Kharagpur Spring Semester, 2022-23

April 28, 2023

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources whenever necessary.

Date: April 28, 2023 (Soumya Ranjan Patra)

Place: Kharagpur (18ME3AI30)

CENTRE OF EXCELLENCE IN ARTIFICIAL INTELLIGENCE

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled “Memory Augmented Neural Network for Extreme Class Imbalance” submitted by Soumya Ranjan Patra (Roll No. 18ME3AI30) to Indian Institute of Technology Kharagpur towards the fulfillment of requirements for the award of the degree of Master of Technology in Artificial Intelligence, Machine Learning, and Applications is a record of bona fide work carried out by him under my supervision and guidance during the academic year 2022-23.

Date: April 28, 2023

Place: Kharagpur

Professor Jiaul Hoque Paik

Centre of Excellence in Artificial Intelligence

Indian Institute of Technology Kharagpur

Kharagpur - 721302, India

Abstract

Name of the student: **Soumya Ranjan Patra**

Roll No: **18ME3AI30**

Degree for which submitted: **Master of Technology**

Department: **Centre of Excellence in Artificial Intelligence**

Title: **Memory Augmented Neural Network for Extreme Class Imbalance**

Thesis supervisor: **Professor Jiaul Hoque Paik**

Date of thesis submission: **April 28, 2023**

In this project, we will study the usage of Memory Augmented Neural Networks (MANN) in various fields. We will propose a model for Visual Question Answering (VQA) inspired by other literature using MANN. Our model will use Transformer for image and text feature extraction. After that, an LSTM-controlled MANN is proposed to train the dataset. The use of MANN in this problem statement is new, and no prior work exists to the best of our knowledge.

Acknowledgments

The completion of this study could not have been possible without the constant guidance and support of Prof. Jiaul Hoque Paik, our respected Project Guide. I am thankful to him for giving me the opportunity to work on this new and exciting topic. I would also like to Prof. Adway Mitra and Prof. Manjira Sinha for the smooth conduction of the project selection process. Finally, I want to convey my regards to my family members and friends who have helped me throughout this time.

Table of Contents

1. Problem Introduction and Motivation

1.1 Extreme Class Imbalance

The problem related to extreme class imbalance is very common in our day-to-day life. For example, in the medical diagnosis field, the number of samples categorized as positive is very likely to be much lower than the number of samples categorized as unfavorable. In the case of mechanical fault diagnosis, the probability of a mechanical failure is also very lower than that of a successful mechanical operation. The potential risk in the case of data imbalance is that learning models tend to favor the majority of samples. As pointed out by [Zhan ao Huang et al., 2022], data imbalance often tends to negatively affect the case of neural networks. This problem exacerbates when data turns out to be highly imbalanced as the class which is in the majority creates a bias towards it in the dataset. Most of the present solutions using neural network approaches to handle this problem heavily rely on rebalancing or reweighting the given data. The motive of these methods is to recover the features of balanced data.

The methods shown in various articles/research papers for handling the extreme class imbalance in machine learning can be classified majorly into the following three categories:- algorithm level, data level, and hybrid approaches. The data level technique uses different data sampling approaches to reduce the imbalance level. These generally include under-sampling, over-sampling, and Synthetic Minority Over-sampling Technique (SMOTE) for handling the class imbalance. Algorithm-level methods are commonly implemented with a weight or a cost schema that includes modifying its output, or the learning parameters, to reduce the bias towards the group in the majority. The hybrid systems methods use some strategies to combine both the sampling and the algorithmic methods.

The biggest problem in the case of data imbalance is that the model cannot remember what might have happened a long time back during its training. LSTM is capable of remembering up to some time, but it does not work well in the case of extremely imbalanced datasets as it also has a limitation of up to when it can remember. That is

why we are trying to come up with an use of external memory for handling this extreme case. We will look into various use of Memory Augmented Neural Networks (MANN) in the science world and then try to devise a solution to our problem.

1.2 Visual Question Answering

Visual Question Answering (VQA) is a task that involves answering natural language questions about an image. The goal of VQA is to create intelligent systems that can understand both the visual content of an image and the semantic meaning of a natural language question. This task requires deep learning models that can process both images and natural language, making it a challenging task.

VQA can be categorized into two main types: open-ended and multiple-choice. In open-ended VQA, the system generates a free-form answer to the question. In multiple-choice VQA, the system provides a set of answer choices, and the user selects the correct answer from the set. Multiple-choice VQA is easier than open-ended VQA because the system only needs to generate a set of answer choices instead of a free-form answer.

VQA has a wide range of applications, such as image captioning, visual search, and image retrieval. VQA can also be used in assistive technologies for visually impaired people, where the system can answer questions about the surroundings or objects.

Several datasets have been created for VQA, including the VQA dataset, for example, the COCO-QA, MSCOCO, and the Visual Genome dataset. These datasets contain images and corresponding question and answer pairs. It can be used to train and evaluate VQA models.

There has been a wide variety of deep learning model implementations for VQA.

- Convolutional Neural Networks (CNNs): CNNs are commonly used as a backbone for VQA models. They are used to extract image features that can be fed into the question-answering module. CNNs can be pre-trained on large datasets, such as ImageNet, to learn generic image features that can be fine-tuned for VQA. Examples of CNNs used for VQA include VGG, ResNet, and Inception.

- Recurrent Neural Networks (RNNs): RNNs are used for processing natural language questions. They can be used to encode the question into a fixed-length vector that can be combined with the image features for answering the question. Examples of RNNs used for VQA include Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

- Attention Mechanisms: Attention mechanisms can be used to focus on specific regions of the image that are relevant to the question. They can be used to weigh the image features based on their importance for answering the question. Examples of attention mechanisms used for VQA include spatial attention, channel attention, and multi-modal attention.

- Memory Networks: Memory Networks can be used to store and retrieve information that is relevant to the question. They can be used to store the image features and question embeddings and retrieve the relevant information for answering the question. Examples of Memory Networks used for VQA include End-to-End Memory Networks (E2EMN) and Dynamic Memory Networks (DMN).

- Transformers: Transformers are a type of neural network architecture that can process sequences of data, such as natural language questions. They can be used to encode the question and generate an answer in a single forward pass.

Examples of Transformers used for VQA include Vision-and-Language Transformers (ViL-T5) and VisualBERT.

- Ensemble Models: Ensemble models can be used to combine multiple models to improve the overall performance. They can be used to combine the strengths of different models and mitigate their weaknesses. Ensemble models can be built by combining multiple CNNs, RNNs, attention mechanisms, and memory networks.

In summary, various deep learning models have been used for Visual Question Answering (VQA). These models include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Attention Mechanisms, Memory Networks, Transformers, and Ensemble Models. Each of these models has its strengths and weaknesses, and the choice of the model depends on the specific requirements of the VQA task.

2. Literature Review

In order to apply Memory Augmented Neural Network (MANN) for extreme class Imbalance Datasets, we first had to understand how MANN actually works. We have gone through various literature reviews and these are the most important of them. We also have gone through the implementation of Visual Question Answering in different literatures. We summarise the literature below as well.

2.1. Neural Turing Machine (Graves et al., 2014)

Neural Turing Machines (NTMs) are defined as an instance of Memory Augmented Neural Networks. It is a new class of recurrent neural networks which uses an external memory unit to separate the computation from the memory. The result of the series of experiments done using NTMs is that it surpasses Long Short-Term Memory (LSTM) in terms of performance. It helps us clearly understand the concept of the attention mechanism. In Fig. 1, the interaction between the read and write head with a controller is shown. Fig. 2 shows a flow diagram of the attention mechanism. It demonstrates the use of Content-based and Location-based attention mechanisms for performing various tasks.

The experimental implementation of this model shows that the model is able to learn basic algorithms like copying and sorting a given dataset. But the selling point of this model was that it could perform those tasks well beyond its training data range. Fig3. below shows that given input data, it was able to copy it successfully. We implemented it and found out to be very accurate. This implementation has been the reference for most of the memory-augmented neural network that was proposed later on in this field. We use the importance of content-based attention as the datasets that we are going to deal with will require more content-based attention and not location-based attention.

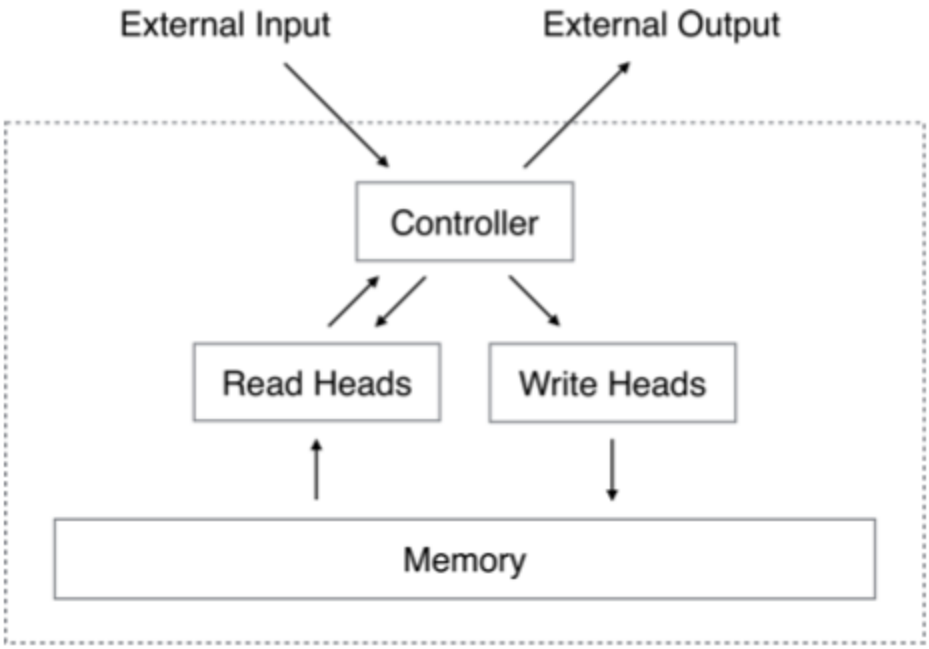


Fig 1. Neural Turing Machine Architecture (Graves et al., 2014)

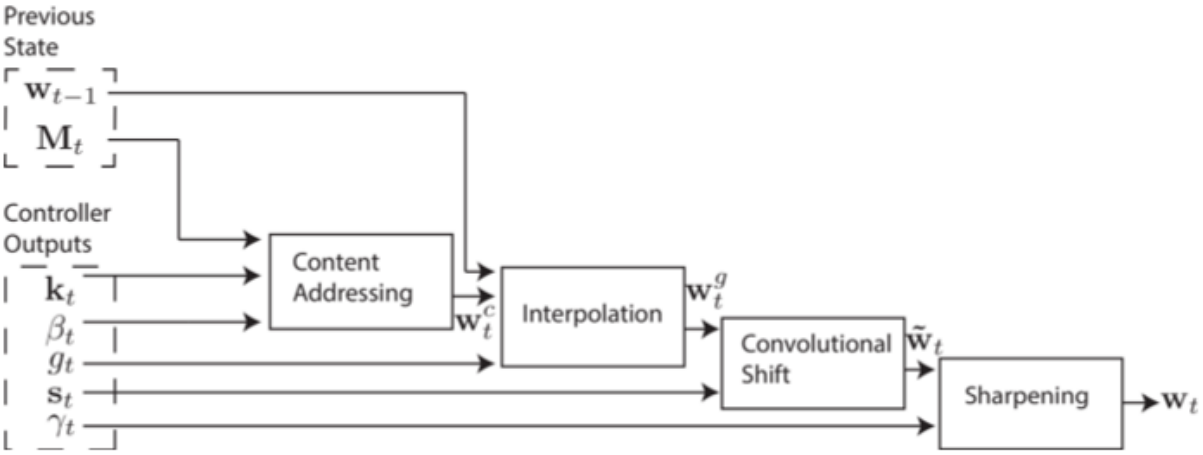


Fig 2. Flow Diagram of the Addressing Mechanism (Graves et al., 2014)

Fig 3a. Input data, **Fig 3b.** Output data

2.2. Few-shot Image Classification (Geethan et al., 2021)

This paper shows the effect of the usage of Memory Augmented Neural Network in case of the presence of a scarce data set. It uses the concept of dividing the dataset into a query set and a support set. The query set updates/writes to the external memory, and the support set accesses/reads from the external memory with a proper attention mechanism.

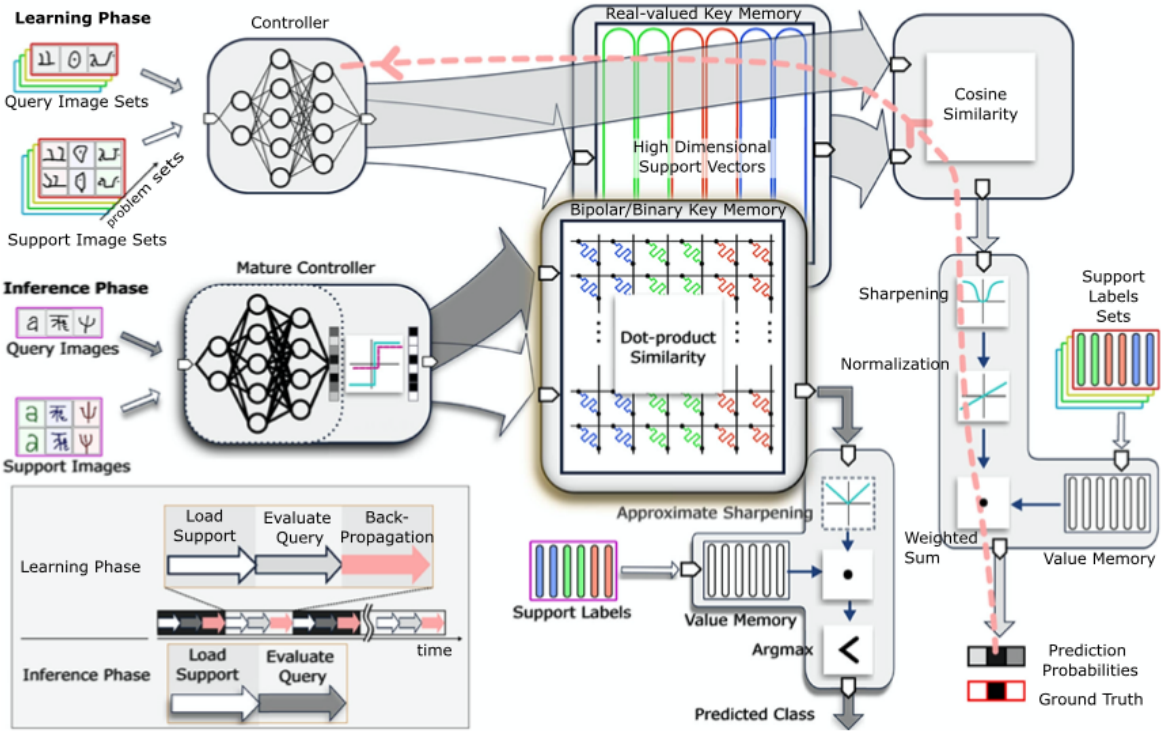


Fig. 4 Proposed robust HD MANN architecture. (Geethan et al., 2021)

In both the learning and interference phases, the data are divided into query sets and support sets. In the learning phase, if the key generated is from the support set, then that vector and corresponding labels are stored in external key-value memory. Values are one-hot labels. If the generated vector belongs to the query set, it is compared to all the keys in the external memory using a similarity mechanism. For attention purposes, the similarities are then transformed into weightings that have a unit-valued norm. The attention mechanism consists of comparison (cosine similarity) and is followed by sharpening.

In case the key is generated from the query set, the similarity will be higher with similar support vectors in the key-value memory. We use the one-hot labels in the value memory corresponding to the keys and take argmax to predict the outputs. In the interference phase, a similar approach is taken. The query set is used to evaluate the predictions, and loss is computed based on the classification errors in the query phase and backpropagation. The model uses a Convolutional Neural Network as a controller.

2.3. Visual Question Answering using LSTM and VGGNet

The model from (Agarwal et al., 2016) uses a bidirectional LSTM to encode all the questions. It uses the last hidden layer of the ResNet or VGGNet to encode the input images. Then the image features are L2 normalized. Both the questions and the image features vectors are transformed into a common space. After that, they are fused via element-wise multiplication. Post which this is passed through a fully connected layer and a softmax layer to obtain the predicted answers finally.

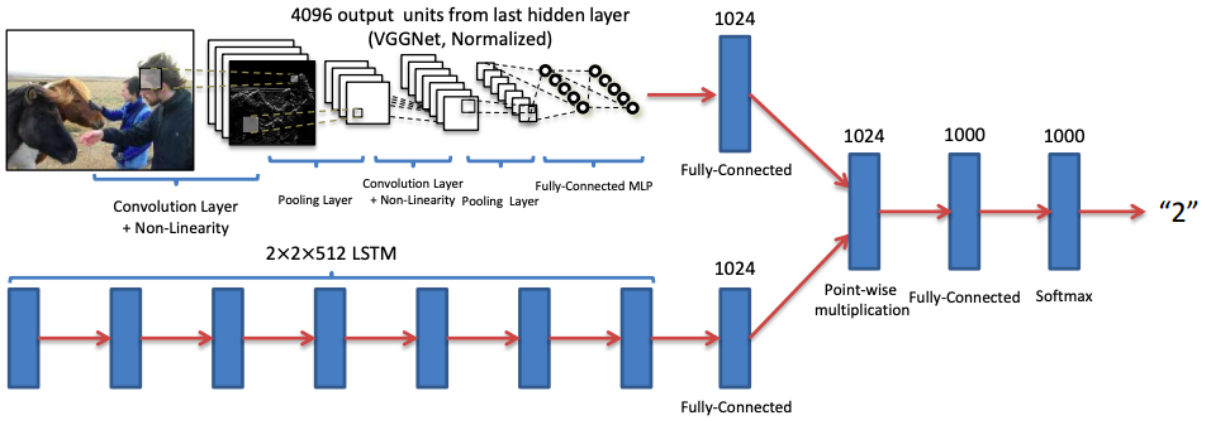


Fig.5 LSTM Architecture for VQA (Agarwal et al., 2016)

2.4 VQA using Memory Augmented Neural Network

This model is based on (Chao Ma et al., 2018) and (Santoro et al., 2016). It uses co-attention for merging the image and question feature and then uses the Memory Augmented Neural Network along with using LSTM as a controller. It shows that MANNs are a good way to maintain long-term memory for scarce training datasets. And it is really important for the case of Visual Question Answering.

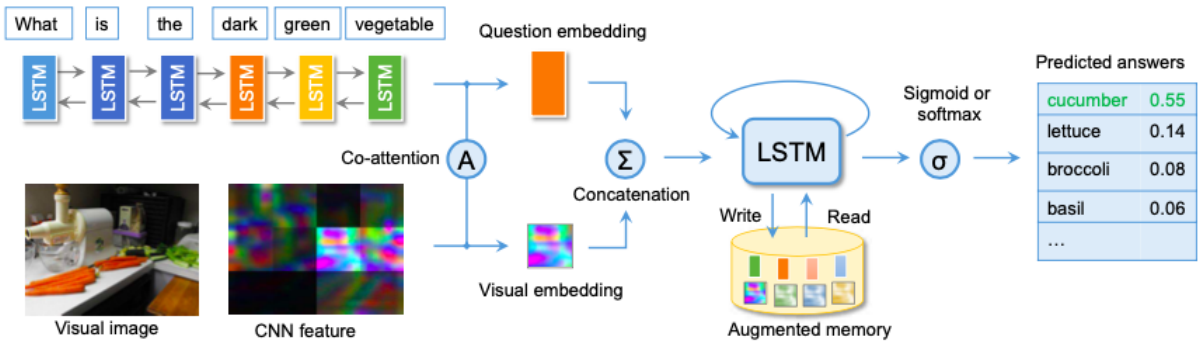


Fig 6. VQA using MANN architecture

2.4.1 Image Embedding

A VGGNet-16 was used for feature extraction. First, the images are resized into 448 x 448 before feeding it into the Transformer. After that the image features corresponding to 14 × 14 spatially-distributed regions have been extracted. The output features can be denoted by $[v_1,...,v_N]$, where N = 196 is the total number of regions and v_n is the n-th feature vector.

2.4.2 Question Embedding

An LSTM was used for preprocessing questions. The questions are tokenized, and embedding is obtained. Further features were extracted corresponding to each word by feeding it to a bidirectional LSTM. The t-th word is represented by q_t, which is a concatenation of hidden states from both directions of the LSTM.

2.4.3 Sequential Co - Attention

The model proposes a co-attention mechanism (Chao Ma et al., 2018) to attend to the most relevant type of features in the image and question feature vectors. Fig.7 represents the visual representation of it. Taking v_n and q_t we can compute a base vector m_0 as shown below:-

$$\mathbf{m}_0 = \mathbf{v}_0 \odot \mathbf{q}_0 \tag{0.1}$$

$$\mathbf{v}_0 = \tanh\left(\frac{1}{N}\sum_n \mathbf{v}_n\right) \tag{0.2}$$

$$\mathbf{q}_0 = \frac{1}{T}\sum_t \mathbf{q}_t \tag{0.3}$$

After that, a neural network of 2 layers was used for the co-attention process.

For visual attention, the final vector is v* and is calculated as shown in the equation below:-

$$\mathbf{h}_n = \tanh\left(\mathbf{W_v v}_n\right) \odot \tanh\left(\mathbf{W_m m}_0\right) \tag{0.4}$$

$$\alpha_\mathbf{n} = \text{softmax}\left(\mathbf{W_h h}_n\right) \tag{0.5}$$

$$\mathbf{v}^* = \tanh\left(\sum_{n=1}^{N} \alpha_n \mathbf{v}_n\right) \tag{0.6}$$

For question attention, the following formulae are used to calculate q*:-

$$\mathbf{h}_t = \tanh\left(\mathbf{W_q q}_t\right) \odot \tanh\left(\mathbf{W_m m}_0\right) \tag{0.7}$$

$$\alpha_\mathbf{t} = \text{softmax}\left(\mathbf{W_h h}_t\right) \tag{0.8}$$

$$\mathbf{q}^* = \sum_{t=1}^{T} \alpha_t \mathbf{q}_t \tag{0.9}$$

where W_v, W_q, and W_h are hidden states. Then concatenate q* and v* to get the final representation of the image and question pair as x_t={v*,q*}

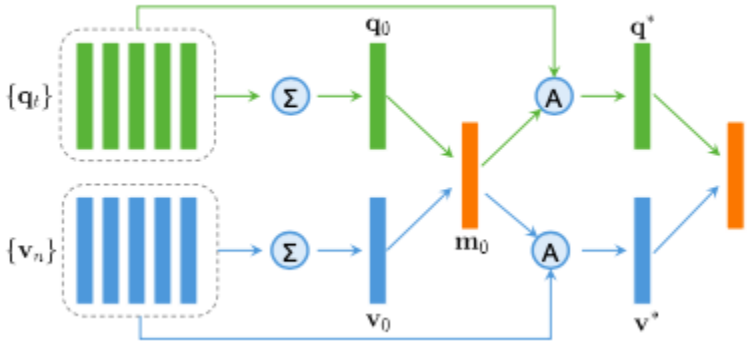


Fig 7. Co-attention Mechanism

2.4.4 Memory Augmented Neural Network

An LSTM controller was used to interact with the external memory. Taking {xt,yt}, where t = 1,..., N, be the overall N training data, where the term x_t denotes the concatenation of visual and question feature vectors, and y_t is the corresponding answer vector which is one-hot encoded. We then feed x_t into the LSTM controller and get h_t as:-

$$\mathbf{h}_t = LSTM(\mathbf{x}_t, \mathbf{h}_{t-1}) \tag{0.10}$$

In order to read from the external memory M_t, hidden vector h_t was taken as a query for M_t. Firstly, cosine distance between M_t and h_t has been computed to obtain the similarity of a query vector and each row in the external memory:

$$D\left(\mathbf{h}_t, \mathbf{M}_t(i)\right) = \frac{\mathbf{h}_t \cdot \mathbf{M}_t(i)}{\|\mathbf{h}_t\| \, \|\mathbf{M}_t(i)\|} \tag{0.11}$$

After that, read weight vector was computed by taking the softmax over the cosine distances we calculated earlier:

$$w_t^r(i) = \text{softmax}\left(D\left(\mathbf{h}_t, \mathbf{M}_t(i)\right)\right) \tag{0.12}$$

The retrieved memory r_t is taken and concatenate with h_t to get final embedding $o_t = [h_t , r_t]$ for obtaining output classifier

$$\mathbf{r}_t = \sum_i \mathbf{w}_t^r(i) M_i$$

In NTM (Gravels et al., 2014), they have taken both memory-based and content-based addressing. Location-based addressing was good for the case where there needed to be long jumps across the memory. This was required for sequence-based data. This data do not need this kind of attention. They only need content-based addressing here. Therefore Least Recently Used Access (LRUA) module was used in the paper, which is a pure content-based memory writer. It is used to write memories to one of the two:

either the least used memory location or the most recently used memory location. This module ensures accurate encoding of the relevant, i.e., the most recent information, and a pure content-based retrieval. The new feature vectors is written either to rarely-used locations in order to preserve the recently encoded information or it is written to the last used location. Writing to recently used locations works as an update to the memory with new and more relevant pieces of information. They have identified which option to choose using interpolation between the previous read weights and weighted scale according to using weights. Then a decaying parameter is used to update the usage weight at each time step, and read and write weights are added to it to get new usage weights, as shown below.

$$\mathbf{w}_t^u \leftarrow \gamma \mathbf{w}_{t-1}^u + \mathbf{w}_t^r + \mathbf{w}_t^w \tag{0.13}$$

A notation m(v, n) is declared, which denotes the n-th smallest element of vector v. The least-used weights for a certain time-step can be calculated as:

$$w_t^{lu}(i) = \begin{cases} 0 & \text{if } w_t^u(i) > m\left(\mathbf{w}_t^u, n\right) \\ 1 & \text{if } w_t^u(i) \leq m\left(\mathbf{w}_t^u, n\right) \end{cases} \tag{0.14}$$

Further, write vector can be obtained using $\sigma(\cdot)$ is a sigmoid function, $e^x/(1+e^x)$, and α is the scalar gate parameter to be able to interpolate between the weights

$$\mathbf{w}_t^w \leftarrow \sigma(\alpha)\mathbf{w}_{t-1}^r + (1 - \sigma(\alpha))\mathbf{w}_{t-1}^{lu} \tag{0.15}$$

Writing to the external memory is done according to the write weights calculated earlier.

$$\mathbf{M}_t = \mathbf{M}_{t-1}(i) + \mathbf{w}_t^w(i)\mathbf{k}_t \tag{0.16}$$

o_t was then passed to generate output distribution. The categorical distribution then obtains a vector p_t whose elements will show each class outcome:

$$\begin{aligned} \mathbf{h}_t &= tanh(\mathbf{W}_o \mathbf{o}_t) \\ \mathbf{p}_t &= softmax(\mathbf{W}_h \mathbf{h}_t) \end{aligned} \tag{0.17}$$

Given the output distribution p_t, in the training stage, the network is then optimized by minimizing the loss over the input one-hot encoded label vector. Loss function:-

$$\mathbf{L}(\theta) = -\sum_t \mathbf{y}_t^T log \mathbf{p}_t \tag{0.18}$$

3. Proposed Model

Our model is similar to the model described in 2.4 with some changes. We have taken BERT(Bidirectional Encoder Representations from Transformers) for getting question embedding and ViT(Visual Image Transformer) for image embedding. Also, we have skipped the co-attention that has been described in 2.4.3. Therefore we need not divide the image into different spatial regions.

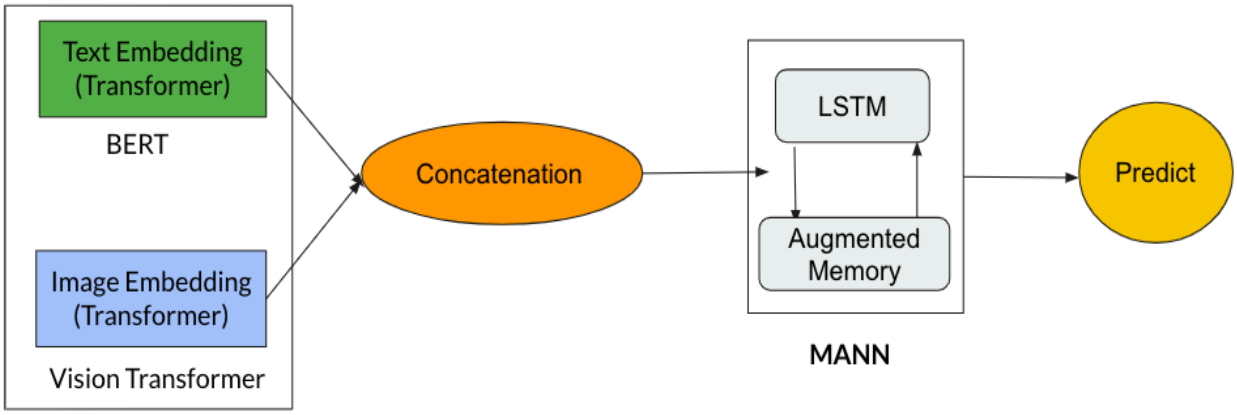


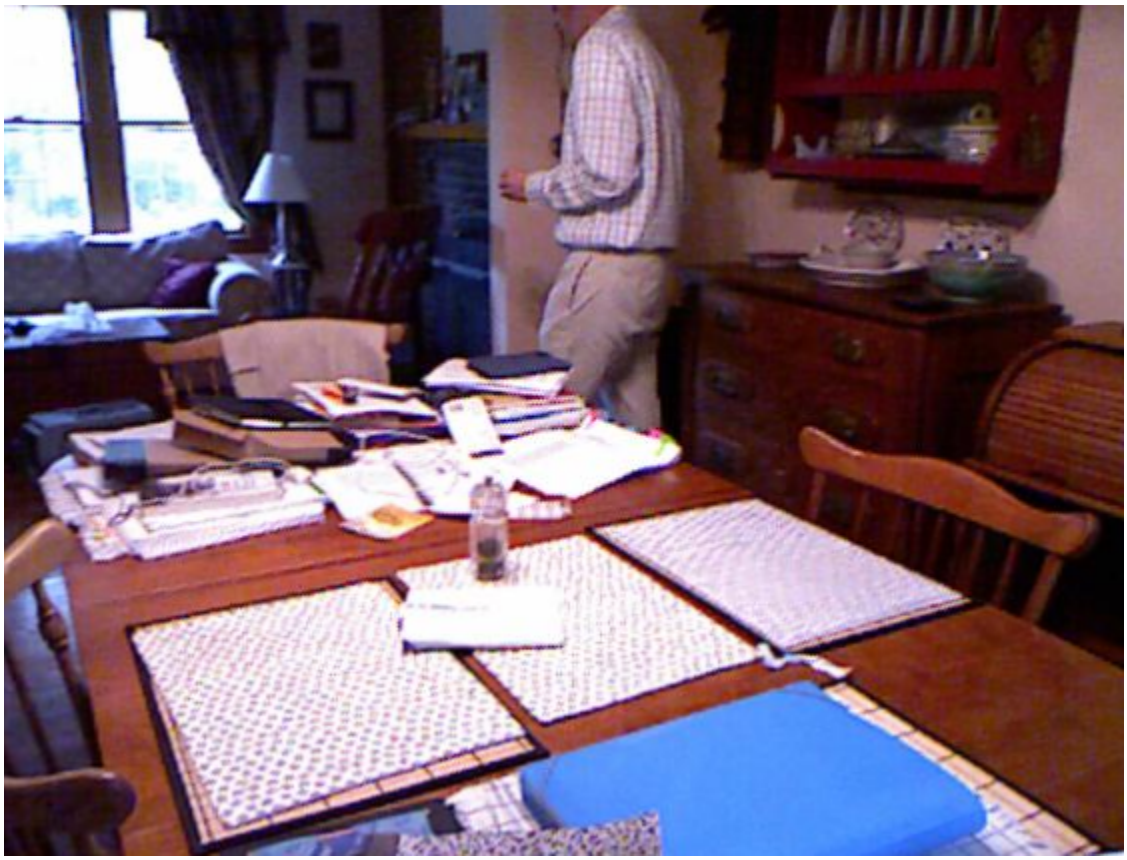
Fig 8. Proposed model Architecture

We have concatenated the image and question feature and fed it to an LSTM-controlled MANN as described in 2.4.4. The output from the controller is taken and passed through a sequential layer consisting of the Linear layer, RELU, and Dropout. After that, we use a linear layer to get the answer. The output dimension of this linear layer is equal to the length of the answer space.

4. Dataset

The DAQUAR (Dataset for Question Answering on Real-world Images) dataset is a well-known benchmark in the field of visual question answering (VQA). It was introduced in 2015 by [Malinowski et al.] to evaluate the ability of models to reason about objects and their spatial relationships in images based on natural language questions. The questions in the DAQUAR dataset cover a wide range of topics, including object identification, counting, spatial relationships, and reasoning about multiple objects. We in our model have used identification and counting portion.

An example of the same is the following:-



Question: What is on the sofa?
Answer: pillow

Fig. 9 Showing an example of VQA

4.1 Dataset Insights

Number of Questions	12468
Average number of words in questions	11.5
The shortest question (in words)	7
The longest question (in words)	31
Total length of answer space	582
Total number of images	1449
Mean Number of questions per image	8.6
the most frequent answer object	table (469 occurrences)
the 2nd most frequent answer object	chair (412 occurrences)
the most frequent answer number	2 (554 occurrences)
the 2nd most frequent answer number	3 (327)

Table 1. Various insights from DAQUAR dataset

5. Implementation

Following are the parameters and models used:-

- Used pytorch for writing the code.
- Used the pre-trained transformer models (“bert-base-uncased” and “google/vit-base-patch16-224-in21k”) models from Hugging Face AutoModel.
- We used AutoTokenizer for tokenizing the questions. The max_length of embedding taken was 24.
- We used AutoFeatureExtractor for getting 224 x 224 sized feature.
- There were more than 19 crore trainable parameters.
- batch size = 16.
- Learning rate= 5e-05
- Optimizer = AdamW
- intermediate_dim=512
- hidden_dim=768
- N = 128, M= 40
- train_size : eval_size = 6:5

Following are the dimension of the parameters associated with MANN

Parameter	**Dimension**
Memory	batch_size x N x M
wtw, wtr and wtu	batch_size x N
rt and ht	batch_size x M

Table 2. Dimension of parameters of model

6. Results

We got the results with and without using MANN in the model. Below are the loss, F1 Score, and accuracy plots for both models. We can see that we are getting slightly better results with the use of MANN because of the additional layer of external memory. We have shown different outputs in Fig. 10 and Table 3 below.

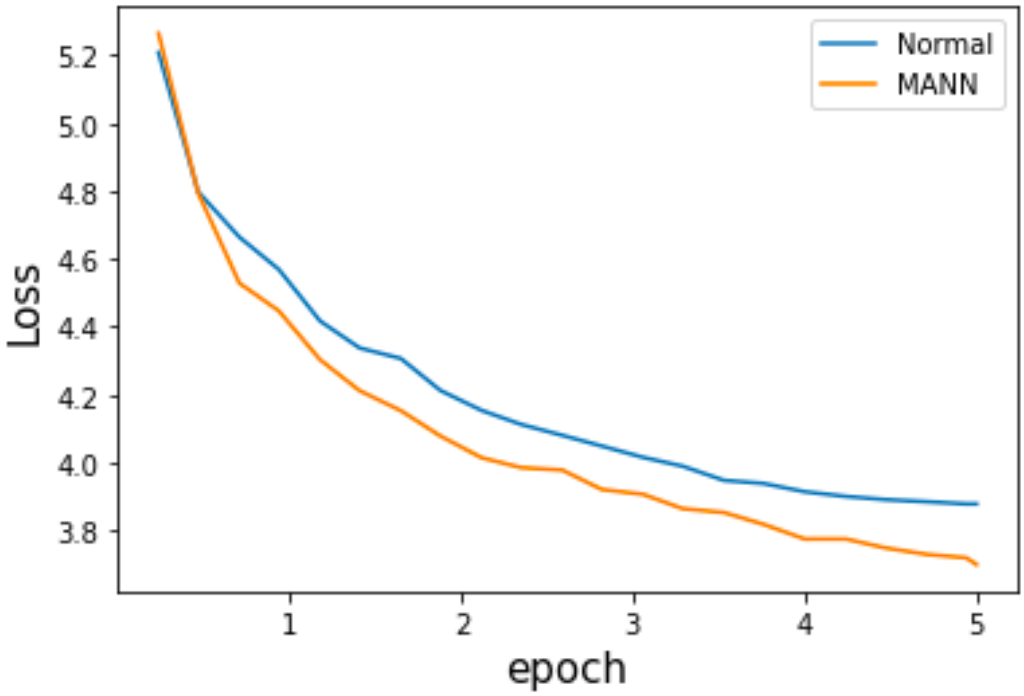


Fig. 10 (a) Comparison of loss over epoch in both models

Although the number of epochs is five but it is evident that the results improve over time.

	Using MANN	Without using MANN
Loss	3.74	3.87
Accuracy	0.21	0.195
F1 Score	0.0185	0.0117

Table 3:- Output of Model using and without using MANN

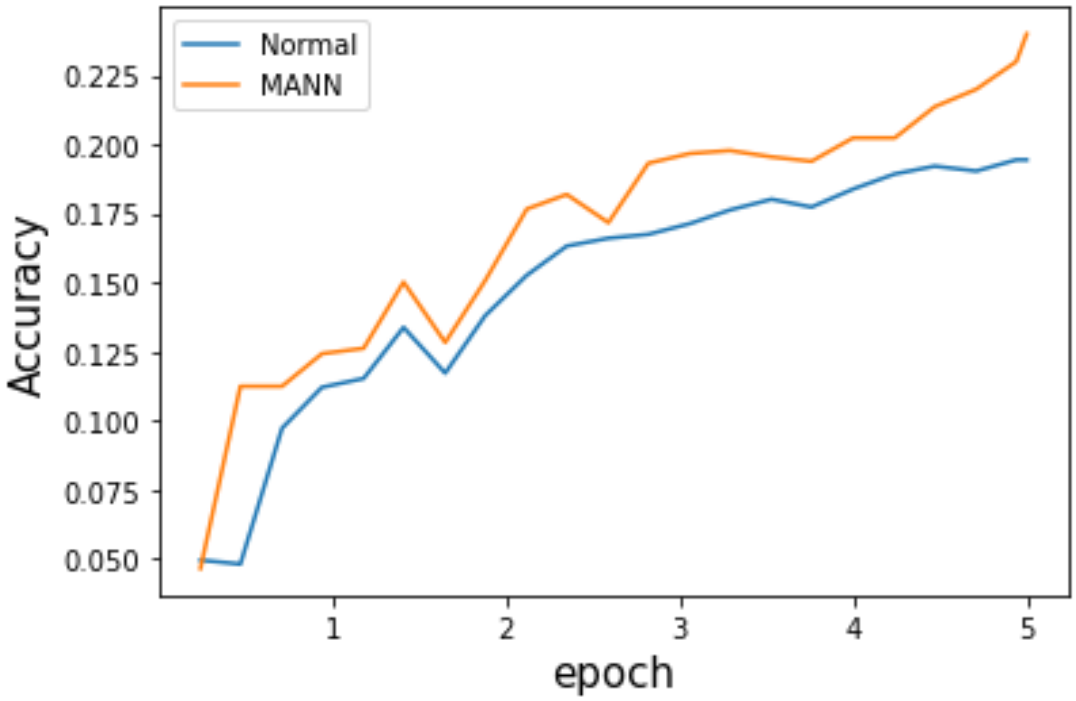


Fig. 10(b) Accuracy over epoch

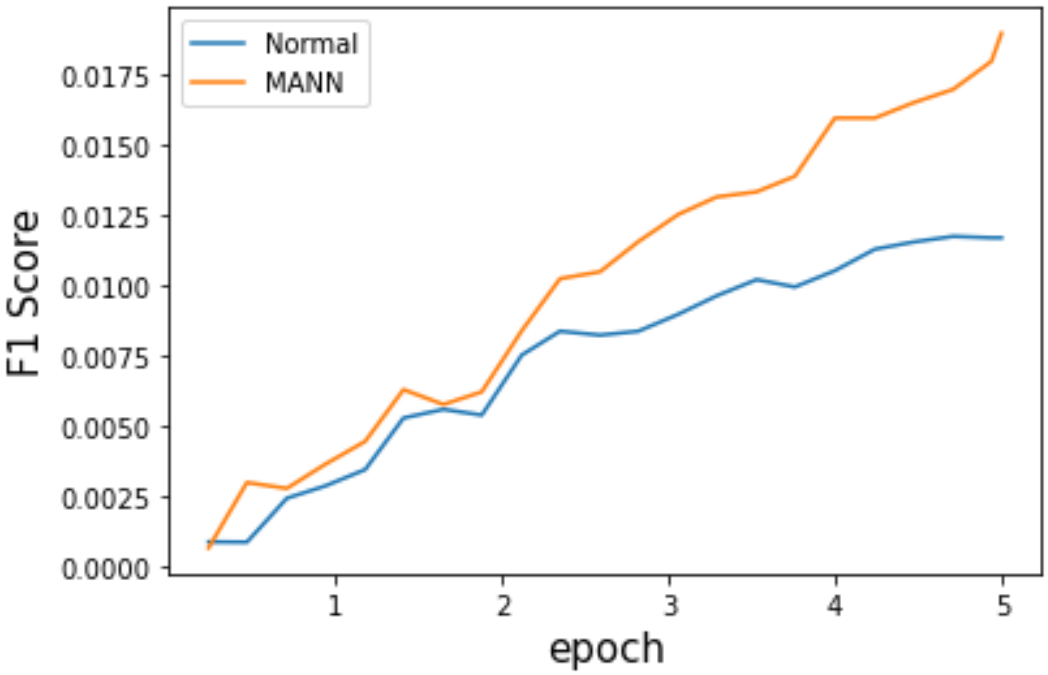


Fig. 10(c) F1 Score over epochs

7. Conclusion and Future Prospects:-

We have implemented a basic version of MANN for Visual Question Answering in this paper. We can extend our work to implement the model described in 2.4, which includes a co-attention mechanism.

We can remove the natural language processing layer and can use the Visual Transformer for feature extraction and an LSTM-controlled memory-augmented neural network, as shown in Fig. 11, for solving problems like disease recognition from images, where we have extreme class-imbalanced datasets. In our model, we have used an LSTM-controlled MANN, but we can also use a feed-forward neural network as a controller.

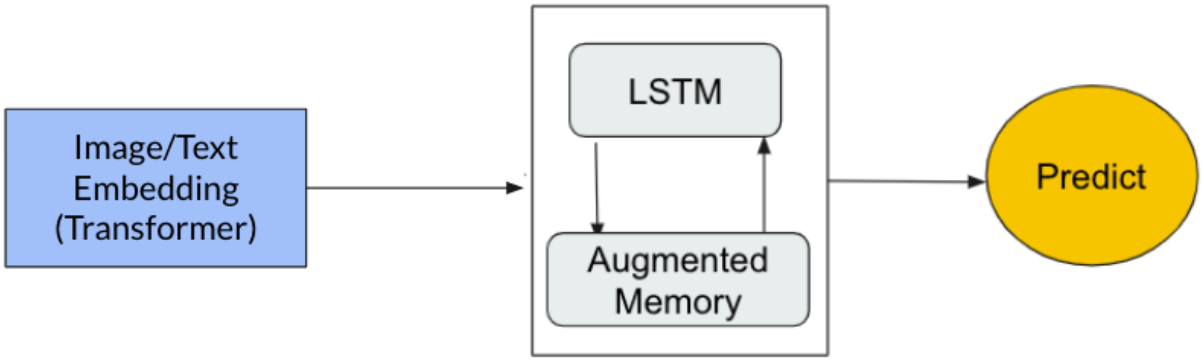


Fig 11. Proposed model for other Extreme Class Imbalance Datasets

8. Bibliography

[S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.]

[Aishwarya Agrawal∗ , Jiasen Lu∗ , Stanislaw Antol∗ , Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, Devi Parikh Visual Question Answering, 2016]

[A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap. Meta-learning with memory-augmented neural networks. In *Proc. Int. Conf. Mach. Learn.*, 2016.]

[Graves, Alex, Wayne, Greg, and Danihelka, Ivo. Neural turing machines. arXiv preprint arXiv:1410.5401, 2014.]

[Johnson, J.M., Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J Big Data* 6, 27 (2019). https://doi.org/10.1186/s40537-019-0192-5]

[Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, Qi Tian 2019, Deep Modular Co-Attention Networks for Visual Question Answering]

[Lalithkumar Seenivasan, Mobarakol Islam, Adithya K Krishna, Hongliang Ren, 2022, Surgical-VQA: Visual Question Answering in Surgical Scenes using Transformer]

[Mark Collier, Joeran Beel,2018, Implementing Neural Turing Machines]

[Zhan ao Huang et al., 2022, A neural network learning algorithm for highly imbalanced data classification]

[Geethan Karunaratne, Manuel Schmuck, Manuel Le Gallo, Giovanni Cherubini, Luca Benini2, Abu Sebastian & Abbas Rahimi, 2021, Robust high-dimensional memory-augmented neural networks]

[Chao Ma, Chunhua Shen∗, Anthony Dick, Qi Wu, Peng Wang, Anton van den Hengel, and Ian Reid Australian Institute for Machine Learning, and Australian Centre for Robotic Vision, The University of Adelaide]

[BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova]

[T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla´r, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, 2014.]

[Ask Your Neurons: A Neural-based Approach to Answering Questions about Images Mateusz Malinowski, Marcus Rohrbach, Mario Fritz, Max Planck Institute for Informatics, Saarbrucken, Germany ¨ 2UC Berkeley EECS and ICSI, Berkeley, CA, United States]