Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 173 (2020) 234–244

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on *Smart Sustainable Intelligent Computing and Applications* under ICITETM2020

Employing Differentiable Neural Computers for Image Captioning and Neural Machine Translation

Rishab Sharma[a*], Anupam Kumar[b], Deepak Meena[c], Sanidhya Pushp[d]

[a]*Fynd, Mumbai, Maharashtra 400093,India*
[b]*Maharaja Agrasen Institute of Technology, Delhi, India*
[c]*NITK Surathkal, Mangalore, Karnataka,India*
[d]*HSBC Bank Delhi, India*

Abstract

In the history of artificial neural networks, LSTMs have proved to be a high-performance architecture at sequential data learning. Although LSTMs are remarkable in learning sequential data but are limited in their ability to learn long-term dependencies and representation of certain data structures because of the lack of external memory. In this paper, we tackled two main tasks, one is language translation and other is image captioning. We approached the problem of language translation by leveraging the capabilities of the recently developed DNC architectures. Here we modified the DNC architecture by including dual neural controllers instead of one and an external memory module. Inside our controller, we employed a neural network with memory-augmentation which differs from the original differentiable neural computer, we implemented a dual controller's system in which one controller is for encoding the query sequence whereas another controller is for decoding the translated sequences. During the encoding cycle, new inputs are read and the memory is updated accordingly. In the decoding cycle, the memory is protected from any writing from the decoding controller. Thus, the decoder phase generates a translated sequence at a time step. Therefore, the proposed dual controller neural network with memory-augmentation is then trained and tested on the Europarl dataset. For the image captioning task, our architecture is inspired by an end-to-end image captioning model where CNN's output is passed to RNN as input only once and the RNN generates words depending on the input. We trained our DNC captioning model on 2015 MSCOCO dataset. In the end, we compared and shows the superiority of our architecture as compared to conventionally used LSTM and NTM architectures.

Keywords: Image Captioning, Deep Architectures, Neural Network, Neural Machine Translation

10.1016/j.procs.2020.06.028

1. Main text

With new emerging neural architectures, with time deep learning is pushing AI to the next level. LSTMs [5] has been the go-to neural architecture when it comes to sequential learning. But recent studies suggest a few more sophisticated architectures, mainly MANNs (Memory Augmented Neural Networks). Our work employs two latest MANN architectures NTM (Neural Turing Machines) [6] and DNC (Differentiable neural computer) [7]. In out work both of these architectures are tested on two tasks, Language Translation, and Image Captioning.

1.1. Language Translation

Machine Translation is an exacting task in which we use large statistical models. These large statistical models are developed using highly subtle language-producing knowledge. Machine Translation examine the use of software to translate from one language to other and is a sub-part of Computational Linguistics.

Neural Machine Translation is a type of Machine Translation in which we use Deep Neural Network for the translation of one language to another. In our approach, we use embeddings for capturing words and other vector representation like continuous space representations for internal states. Unlike phrase-based statistical models, our model predicts one word at a time using only a single sequence model called Sequence Modelling. The word sequence modeling is done using a recurrent neural network (RNN). However, RNN has two limitations that are the fixed-length problem and Gradient Vanishing/Exploding. To handle these failing basic RNN cell is replaced by Gated Recurrent Unit (GRU) or Long Short-Term Memory (LSTM) cell and then hyperbolic tangent activation is exchanged by Rectified Linear Units (ReLU), the encoder-decoder model is adopted. An encoder (a bidirectional RNN), is used by the neural network to encode a source sentence for a decoder (a second RNN), that is used to predict words in the intent language. The one hidden state was not enough. To further improvise the result in general Attention Mechanism is used. Attention Mechanism stopples a context vector into a gap between decoder and encoder. Attention is a vector and with the help of SoftMax function it frequently the outputs of a dense layer. The input of the context vector is the output of all cells that are used to compute the probability distribution of source language words for each single word decoder wants to generate. It enables the translator to pay particular attention to local or global features.

The Differentiable Neural Computer (DNC) [7] is a neural network architecture with a memory augmentation and the attention mechanism both at the same time. It has two components a controller and a memory. It is an extension of Neural Turing Machine (NTM) [6], with additional features of memory attention to control where memory is stored and temporal attention to store order of events. It is better than some of its predecessors such as the LSTM because it performs tasks that have longer-term dependencies. As the name suggests, DNC is differentiable end-to-end that makes it possible to optimize them in a well-organized and competent way using gradient descent. The DNC model is a Turing complete because of the re-sizability of the memory.

1.2. Automatic Description Generation

Our second task, Image captioning, also known as automatic description generation is a challenging computer vision task as well as a language translation (NLP) problem. It’s a sub-problem of a larger task in computer vision i.e. scene understanding. Recent advances in deep learning have pushed Computer Vision to a new climax by enabling high accuracies in task [8] such as Object Detection, Action Detection, Life Logging, etc. Tasks such as Image captioning demonstrate the potential of deep learning architectures.

To tackle this problem, initially algorithmic approach [9] was taken using similarity measures between images and words. But, in time deep learning entirely changed the process by introducing robust data-driven models which were able to learn the correlation between the spatial representation of images and the sequential patterns in sentences.

Now Automated Image captioning is done using sequence-to-sequence models [2], [10]. The Recurrent neural network learns the semantics of the language used in describing the images, while convolutional neural networks give them the ability to produce words/sentences that are relevant to the image by providing features or by directly passing embeddings. Jointly they act together to form a larger architecture to describe an image, which intuitionally appears obvious. Individual advances in these modules (LSTMS etc.) has improved the capabilities of the final architectures greatly.

Generally, the images are pre-processed and then passed through a CNN. After which the architecture can be either end-to-end [2] or the features could be extracted from CNN and then passed to a different RNN based architecture as described here [10]. Different variations in architecture lead to variations in results. The RNN part is provided with the word-embeddings, which can either be learned or directly generated from models like word2vec [11] which can provide word embeddings. After training the model on a decently large dataset, Captions are generated word by word and each word from the result is appended to the input for next word (naive approach). Generally, the performance of these models is significantly inaccurate, thus heuristic searching techniques such as beam search are often used.

2. Related Works

Image Captioning and Neural Machine Translation Models: Image Captioning is the process of an automatically spawning textual content of an image which is totally based on the objects and actions in the image. In Neural Machine Translation, the encoder (a bidirectional RNN), is used by the neural network to encode a source sentence for a decoder (a second RNN), that is used to predict words in the intent language. But in new approach, [2] the encoder RNN is replaced by CNN (deep convolution neural network). Basically, CNN's embedded the input image to a definite-length vector. Therefore, for image classification, CNN is used as an image “encoder” and then the output of the last hidden layer as an input to the “decoder” (RNN) that spawn sentences. This model is named as Neural Image Caption (NIC). First, this end-to-end system, the neural net is trained using Stochastic Gradient Descent. Second, this model merges sub-networks for language as well as vision models. Finally, NIC beats current state-of-art on many datasets like Pascal dataset, Flickr30K, SBU. On Pascal dataset, BLEU score of NIC was 59 as compared to the state-of-art was 25 and on Flickr30k it improved the earlier result from 56 to 66.

Generally, LSTM is used for sequence modelling but it has some limitations. The new approach was introduced to overcome LSTM's limitations.[3] This approach took the inspiration from the DNC (Differentiable neural computer) and added an extension to the LSTM architecture. The implementation of a DNC is a non-trivial task and also DNCs are difficult to train, instability throughout the learning process, take a long time to converge and prone to over-fitting. The copying task is performed on the bAbI dataset using DNC and DNC performance was better than or equal to LSTM standard [4]. The new version of LSTM called as “MC-LSTM”, where ‘MC’ stands for ‘memory control’. Just like conventional input, forget and output gate they added a new gate which is responsible for reading and writing from the memory matrix (Analogous to the access module of DNC which is responsible for reading and writing from memory) [3]. In original DNC paper [5], the controller can be any neural network but, in this work, the LSTM cell can be thought of as a controller itself and memory are handled by the control gate. Unlike the normal DNC (in case of a recurrent controller), the controller’s hidden state isn’t passed to the next state of the network but a different control vector is sent. They tested the new architecture on an internal Fan Filter Unit dataset and “MC-LSTM” has given better results than the LSTM.

There has been new approach introduced for sequence-to-sequence prediction modelling using a differentiable neural computer to predict a sequence of treatment plans for the future by observing previous digital medical record of a patient [1]. The prediction model is built on the theory that an effective recommended treatment plan shows a clear long-term dependency from the previous record of health information. This problem is approached with two key moderation that is treatment recommendations for a particular subject as a sequence-to-sequence problem for prediction and writes protected policy is applied to the decoding controller. The first controller is for encoding the history and second controller is for decoding the treatment sequences for treatment recommendation. While encoding, the external memory goes through a updation cycle as soon as the new input is read and during the decoding phase, unlike encoding the memory is write-protected. This dual controller writes protected memory-augmented neural net (DCwMANN) model was trained on MIMIC- III dataset and it performs better than existing methods of treatment recommendation.

3. Architecture

3.1. Overview

A Differentiable Neural Computer [7] is an end-to-end differentiable neural network system which is coupled to an external matrix for memory. The DNC behavior is independent of the size of the memory unless the memory is saturated or filled to its complete capacity, which is the reason why the memory is termed external memory. The DNC system's network is referred to as the controller, whereas the memory of the DNC can be thought of like the RAM (Random Access Memory), thus making a differentiable CPU which learns its operations using gradient descent. The neural memory frameworks [12] [13] recently suggested are different from the differentiable neural computer architecture because it allows iterative memory content modification by selectively writing and reading to the memory. The neural Turing machines which are the earlier form of DNC have a similar structure with a different memory addressing mechanism and practically un-expandable thus, a limited memory space.

A differentiable neural computer uses a differentiable attention mechanism to determine the distribution of the rows (locations) in a memory matrix, unlike the conventional non-differentiable computers which use a simple unique address to access the contents of the memory. The weightings distribution shows involvement[AK1] degree of a particular location in a Write and read operation. The memory M is read over by a read vector (R_e) using a read weighting W_e^{Re}. The read vector can be represented by the following equation:

$$R_e^{W_e} = \sum_{i=1}^{N} M_e[i,.]W_e^{R_e}[i] \qquad (1)$$

where the'.' denotes all j= 1...,

In the same way, the write operation first erases the memory with an erase vector e and rewrites it by adding a write vector v using a write weighting.

$$M_e[i,j] \leftarrow M_e[i,j](1 - W_e^{W_e}[i]e[j]) + W_e^{W_e}[i][j] \qquad (2)$$

The weightings are applied and determined by two types of functional units called write and read heads. We have illustrated the operations of the read and write heads in Fig.4.

3.2. Head and Memory Interaction

In DNC, three different types of differentiable attentions are used by the heads. First, differentiable attention is a content lookup in which the controller emits a key vector which is compared to the memory content at that particular location using a cosine similarity measure. This addressing technique is common in both DNC and NTM, it's inspired from content-addressing technique used in Hopfield Networks [14]. The scores generated from the similarity measure are used to determine whether a weighting can be used by the write heads for modifying the existing memory vector or by the read heads for associative recall. It is important to note that a key that matches the memory content partially can be utilized to attend certain operation for that particular location. This property acts as a form of pattern completion, as the content of one address effectively encodes the other address's references, thus a rich mechanism is provided by key-value retrieval for navigation of various data structures associated in external memory.

The second differentiable attention applied in DNC is used to record transitions between consecutively written memory locations in temporal link matrix **L** of NxN dimensionality. L[m,n] approaches 1 if m was the next written location after n, else it falls close to 0. The operation termed as L_w is used to smoothly shift the focus forward towards the memory locations written after the ones emphasized in w, whereas $L_t W$ is used to shift the focus backward. This attention mechanism gives a differentiable neural computer the native ability to remember back a sequence in the same sequential order in which it was originally encoded, even when the encoding phase doesn't perform consecutive writes in adjacent time steps.

The third differentiable attention used in DNC enables the allocation of memory to perform the write operation. The memory usage for each location is represented by a decimal falling between the range 0 and 1, and an unused location picked out by a weighting is delivered to the write heads which automatically increases with each write to that particular location and after each read it gets decreased. This enables the DNC controller to perform reallocation of memory that is freed and is not required any more by the subsystem. This allocation mechanism is independent of the memory content and size which enables a differentiable neural computer to tackle a task by utilizing only one size of external memory and upgrade it to a relatively bigger slot later without undergoing retraining. This makes it possible for a DNC to use a external memory without any bounds by increasing the locations every time a certain minimum threshold usage is passed for any location.

The discussed differentiable attention mechanism's designs are mainly motivated by computational consideration. The content lookup mechanism enables the data structures to be associative, whereas the retrieval of input sequences is enabled by temporal links. The third mechanism allocates the unused locations to the write heads. However, a parallelism is observed between the mammalian hippocampus and the differentiable neural computers on memory mechanics. The memory modification of a differentiable neural computer is one shot and fast like the associative long-term potentiation in the mammalian hippocampal CA1 and CA3 [15]. The DNC takes inspiration from the dentate gyrus, the region of the mammalian hippocampal known for supporting neurogenesis [16]. It helps to increase the representational sparsity in the external memory, thereby enhancing the usage-based memory allocation, memory capacity [17], and weighting sparsity. The increased probability of item sequence recall is also demonstrated by the human free recall experiment, which shows that it has the same order of item recall as the phenomenon dependent on the hippocampus reported by the temporal context model [18], thus showing clear similarity to the formation of temporal links.

3.3. Neural Turing Machines

Another architecture used is NTM (Neural Turing Machine). DNC being NTM's successor is also a MANN architecture. Besides DNC's attention mechanisms and infinite memory, everything between these architectures is common. Neural Turing Machine as its name suggests, takes inspiration from the Turing Machine. Here controller, analogous to the Turing machine, can be thought as the finite state table. Just like the Turing machine, the memory in the NTM is infinite. But, in practice and implementation the memory used is fixed at the when we define the computation graph, unlike the DNC which can extend its memory.

Reading:

The read head produces normalized read weights (1 x N), which are then used for reading the content from the memory. Each row is multiplied with its corresponding weights and then resulting rows are added element-wise to yield the read vector (1 x M). The memory matrix consists of N rows M column each.

$$\sum_t (i) = 1,0 \leq w_t(i) \leq 1, \forall i r_t \leftarrow \sum w_t(i) M_t(i) \tag{3}$$

Writing:

NTM's writing operation is inspired from LSTM's input and forget gates. As write head yields the write weights wt (1 x N) and the erase vector et with M values between 0 and 1. Each row of new memory matrix is acquired by following erase and write operations.

$$\overline{M}_t \leftarrow M_{t-1}(i)[1 - w_t(i) e_t] \tag{4}$$

First, element-wise multiplication of the previous rows and the multiplication of erase vector and elements of the write weights wt subtracted from a row-vector, containing only 1s is acquired (M_t).

Along with erase vector write head yields write vector at which with write weights are used to perform add operation to the memory matrix. Just as in the given eq (5):

$$M_t \leftarrow \overline{M}_t + w_t(i) a_t \tag{5}$$

Addressing Mechanism:

The addressing in NTM is different from the DNC. Along with the content-based addressing mechanism which is common with DNC, location-based addressing is also used in NTM. Location based addressing consist of three steps Interpolation, Convolutional shift, Sharpening. Interpolation decides how much of the new weights to add and how much of old weights to write. The Convolution shift is used for determine if and how much to rotate the weighting. Finally, sharpening is applied to get the final weights.

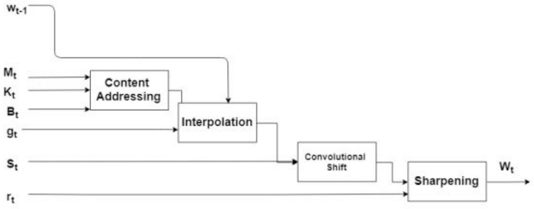

Fig. 1. Flow Diagram of the Addressing Mechanism.

4. Proposed Model for Image Captioning

We present Neural Network Architectures for image captioning while making use of two recent MANN's i.e. NTM and DNC. What sets these architectures apart are their memory and new abilities, while beating the LSTMS in tasks such as copy, repeat copy and question-answering (bAbi dataset) [7] [4] which requires a long-term memory, they're also able to do the conventional tasks that conventional RNNs like LSTMs can do.

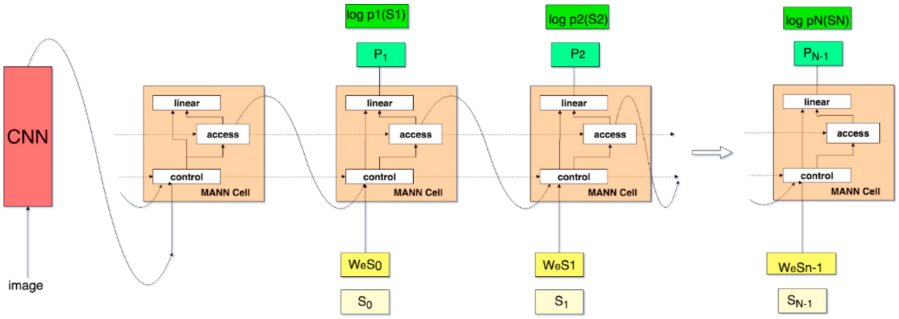

Fig. 2. The CNN model passes the embeddings to the RNN. The access module is responsible for reads and writes from the memory matrix M. The Linear module is feedforward, it takes the output of both controller and access module and applies activation giving the final output at any timestamp. We represent the embedding weights. West represents embeddings of the t-th word in the example sentence.

The architecture presented is inspired from a recent work [2], which presented an end-to-end image captioning model where CNN's output is passed to RNN as input only once and the RNN generates words depending on the input. The input image is given to the CNN which yields image embeddings. These embeddings are then passed to RNN, which encodes the image embeddings and passes them to the next state of RNN. From this state, sentence input is fed. The captions in the dataset are pre-processed which helps to create a dictionary of unique words, these words are one hot encoded, then the embeddings of these words are passed to the RNN. Their study empirically shows that passing the image only once, yields better result as LSTM isn't able to over-fit on the images and their noise this way.

The above Fig.2 depicts the proposed architectures. The CNN doesn't interact with the Memory, but the RNN is assigned as a controller. The controller is LSTM based but could be any type of RNN, in theory, it could even be feedforward, but RNNs are more suitable for sequential tasks. The access module is a generic representation of memory interaction for any type of MANN i.e. in case of DNC it can be replaced by DNC's read/write operations and memory and the same goes for NTM. This module yields the output from the memory, it is responsible for the reading/writing the contents from/in the memory. This provides a certain level of abstraction from the memory interactions, which differ greatly in the two MANNs (DNC and NTM). Everything other than the memory interaction is the same in both DNC and NTM thus both architectures are represented above in the same figure.

At every timestamp, the output of the access module from the previous step is passed to the controller, and the output of the controller is passed to the access module. Both controller and access modules pass their respective states to the next timestamp. At first image, embeddings are acquired from CNN. For the very first timestamp of DNC/NTM, the access module's output which is required in controllers' input is initialized with zeros. Which then is appended to the output of the CNN. The controller now processes the input, returning its state (controller is assumed to be an RNN) and produces the first output. Starting the next timestamp sentence as word embeddings is fed to the RNN. At time t the probability of the t-th word is produced by the taking SoftMax of MANN's output. Using categorical cross-entropy the loss is back propagated in the entire network. It can be minimized with any variant of gradient descent.

4.1. Inference

Descriptions can be generated using various methods, the easiest one is the **Naive approach**. In a naive approach, one word is generated at a time by choosing the word with maximum probability from the output. It is called naïve as no heuristic function is used. Newly acquired words are appended to form an incomplete sentence, these incomplete sentences are passed repeatedly unless the model returns an End-of-Sentence token at that point we've got the complete sentence and no more steps are to be taken. Another famous and preferred method is **Beam Search**. It is a heuristic search technique where the heuristic function is the probabilities acquired from the model. It explores not one, but k possibilities. It helps us tackle description generation as a graph problem, where the destination is the sentence with maximum probability. We have to find the path with maximum probability from a set of k paths.

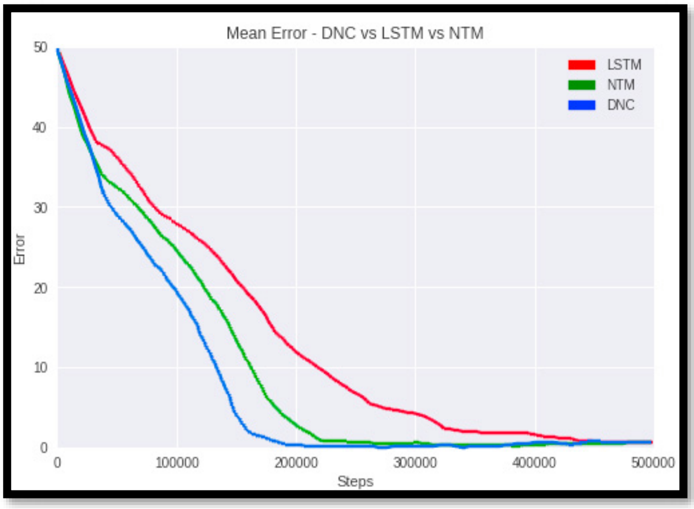

Fig. 3. Learning curves. Median error on 10 training runs (each) for a DNC, NTM and LSTM.

5. Proposed Model for Language Translation

In this section, we demonstrate one of our major contribution to solving the task of language translation by using a deep neural architecture. The architecture is called DCw-MANN [1] or Dual Controller Write-protected Neural Network with Memory Augmentation. Our suggested architecture introduces two contributions which are simple yet vital to the original differentiable neural computers. The first contribution stated in our following study is the use of dual controllers to do the processes of decoding and encoding the sequence to sequence data. The second contribution is the application of a write-protection policy while the memory augmented neural network is in the decoding phase. The encoding phase starts with the input sequence being fed to the embedding layer **WE**, whose output is fed to the encoder which is the first controller (ULSTMe). The first controller reads and writes to the memory at each time step. This information is later used during the decoding phase by the second controller. In the decoding phase, the decoder (second controller) receives the states of the first controller and decodes the incoming encoded vector to its respective embedding.

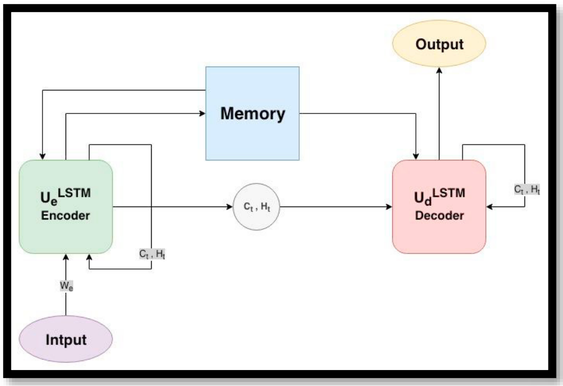

Fig. 4. Dual Controller Memory Augmented Neural Network with write protection. ULSTMe represents the encoding controller instrumented as a LSTM. ULSTMd represents the decoding controller.

The use of dual controllers has a marginal advantage over the use of one controller as it is harder for one controller to learn more than one strategies in the same time frame. Thus the dual controller use makes the learning easy and more focused as compared to a single controller. It is also important to note that the decoder in our architecture can utilize of the previously predicted values from the last embedding layer (WD) as an input combined with the values picked up from the external memory. Next important feature of memory augmented dual controller architecture is its ability to enable the write-protection mechanism during the decoding phase. This feature gives the network a marginal advantage during the decoding step over the writing strategy employed in the vanilla differentiable neural computer, as no fresh input is given to the neural system. Also, some code dependencies are left in the output sequences. These dependencies which are contained among the sequences at the output end are not too long, so the cell memory can properly capture them Ct within the second controller (decoder LSTM ULSTMd). Therefore, the second controller employed in the DNC architecture is inherently prohibited from writing any information to the external memory. In specific words, at a particular time step TS(t) + 1 we have the hidden cell memory as well as the hidden state of both the controllers as:

$$H_{t+1}, C_{t+1} = \begin{Bmatrix} U_e^{LSTM}([W_E v_d, r_t], h_t, c_t); & t \leq L_{in} \\ U_d^{LSTM}([W_D v_d, r_t], h_t, c_t); & t > L_{in} \end{Bmatrix} \tag{6}$$

where vdt is the input sequence code's one-hot encoded vector at a particular time step $t \leq Lin$ and vpt is the one-hot encoded output vector from the decoder at particular time step t > Lin, defined as vpt = onehot(ot), i.e.,:

$$v_{p_t} = \begin{Bmatrix} 1; & i = argmax(o_t[j]) \\ 0; & otherwise \\ & \end{Bmatrix} \tag{7}$$

We suggested to enable the write-protected mechanism as new memory update rule:

$$M_t^e = \begin{Bmatrix} M_{t-1}^e \cdot (E - w_t^W e_t^w) + w_t^W v_t^T; & t \leq L_{in} \\ M_{t-1}^e; & t > L_{in} \end{Bmatrix} \tag{8}$$

where E is a ones matrix of N X D dimensionality, $wtw \in [0,1]N$ represents the write-weight, $et \in [0,1]D$ represents an erase vector, $vt \in RD$ represents a write vector, '.' shows a point-wise multiplication operation, and Lin is the input sequence length.

5.1. Sequence to Sequence translation experiment

In the first experiment, we investigated the language translation capacity of the dual controller differentiable neural computer. We compared our DNC's performance with other neural architectures using the Europarl Machine Translation Dataset [19]. The Europarl is a standard statistical machine translation dataset used for neural machine translation. The dataset comprises of the proceedings that were recorded in the European Parliament. These proceedings are translated into eleven languages from the transcriptions of speakers at the European Parliament. The dataset is mainly a collection of the proceedings of the dating back to 1996. The transcription corpus comprises of about thirty million words for each of the eleven official languages recognized by the European Union. The European Parliament website provides this data corpus in a downloadable format. The Europarl dataset was created by Philipp Koehn, who is known for his book on the topic of statistical machine translation. The dataset has been open sourced for researchers on the Europen Parliament website by the name of European Parliament Proceedings Parallel Corpus 1996-2011. This dataset is often used as a part of machine translation challenges. The latest version of the dataset is version 7, which was released in 2012 and comprised of data from 1996 to 2011.

In our experimentation, we found that when a single differentiable neural computer is trained on ten thousand instances of the Europarl data, a mean error of 3.2 % was achieved during testing, compared to the benchmark of 7.5 % mean error [1]. We also recorded that our dual controller DNC performed much better than both LSTM's [5] and NTM's [6]. LSTM is long short-term memory cell which is the benchmark neural network architecture applied to the many sequence to sequence processing tasks, whereas NTM is short for a Neural Turing Machine. Unlike the previously recorded benchmarks on the mentioned dataset, the given input to our model was single token encoding certain words without any prior preprocessing or any other sentence-level feature extraction.

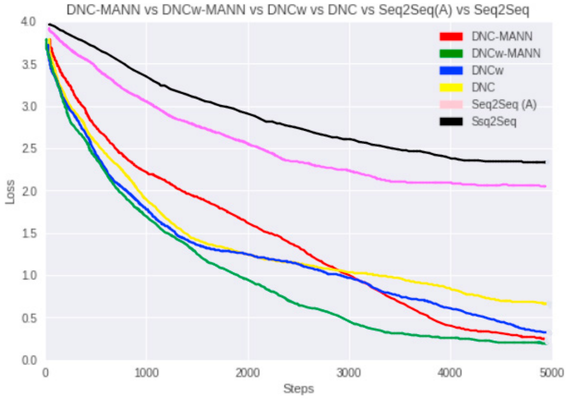

Fig. 5. Training Loss for Neural Machine Translation Task.

6. Conclusion

In the work demonstrated in this paper, we attempted to solve two tasks of image captioning and neural machine translation respectively. For tackling the image captioning task our architecture is inspired by an end-to-end image captioning model where CNN's output is passed to RNN as input only once and the RNN generates words depending on the input. The DNC implementation of our architecture is compared with NTM and LSTM's performance on the same task and the respective mean errors are graphed and shown in the image captioning section. For the second task of neural machine translation we used a dual controller memory augmented neural network with write-protected memory during the decoding cycle. To calculate and compare the neural network's performance on sequence recall, we used NLD - Normalised Levenshtein distance. The normalized edit distance is used to measure the model's performance. The Levenshtein distance is in between the generated and target sequence and is normalized over the length of the longer sequence. The smaller the NLD value denotes a good predicted sequence. The measured values form different models are summarized in Table 1.

In our network design, the order dependencies are preserved between each admission, which allows the use of this sequential information using memory-based methods for showing a better performance. Our work differs from other approaches because we attempted to implement memory augmented network to image captioning and neural machine translation. Our results have shown that our modification of using a dual controller architecture for sequence to sequence processing are really effective for solving such real-world problems. The network architecture implementation can be generalized to all long-term dependency tasks that process sequences to make predictions.

Table 1. Normalized Levenshtein Distance to Target Sequences.

Model	NLD
DNC	0.255
DNCw	0.250
DNC-MANN	0.159
DNCw-MANN	0.085
Seq2Seq	0.712
Seq2Seq (Attention)	0.634

About BLEU: BLEU [28] score is used as the evaluation metric, despite having some drawbacks BLEU is in itself a standard metric and is commonly used in machine translation tasks. It gives n-gram precision between two sentences. Due to limited resources and demanding architectures, it is out of scope to go through large set of hyperparameters and long training durations. Thus, we stopped the training after achieving satisfactory results.

Table 2. Evaluation matric for image captioning task.

Metric	LSTM	NTM	DNC
BELU 1	0.675	0.719	0.729
BELU 2	0.506	0.538	0.557
BELU 3	0.391	0.397	0.409
BELU 4	0.247	0.291	0.312

References

[1] Hung Le, Truyen Tran and Svetha Venkatesh, "Dual Control Memory Augmented Neural Networks for Treatment Recommendations," 2018.

[2] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, "Show and Tell: A Neural Image Caption Generator," 2015.

[3] Itamar Ben-Ari, Ravid Shwartz-Ziv, "Sequence Modeling using a memory controller extension for LSTM," 2017.

[4] C. Hsin, "Implementation and Optimization of Differentiable Neural Computers," 2018.

[5] Sepp Hochreiter, Jurgen Schmidhuber,"LONG SHORT-TERM MEMORY," 1997.

[6] Alex Graves, Greg Wayne, Ivo Danihelka, "Neural Turing Machine," 2014.

[7] Alex Graves, Greg Wayne, Malcolm Reynolds, "Hybrid computing using a neural network with dynamic external memory," 2016.

[8] Bambach, Sven, "A Survey on Recent Advances of Computer Vision," 2013.

[9] Jia-Yu Pan, Hyung-Jeong Yang, C. Faloutsos, P. Duygulu, "GCap: Graph-based Automatic Image Captioning," 2004.

[10] Marc Tanti, Albert Gatt, Kenneth P. Camilleri, "Where to put the Image in an Image Caption," vol. 2, 2018.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, "Distributed Representations of Words and Phrases," 2013.

[12] O. Vinyals, M. Fortunato and Jaitly, "N. Pointer networks. In Advances in Neural," Neural Information Processsing Systems, vol. 28, 2015.

[13] J. Weston, S. Chopra and A. Bordes, "MEMORY NETWORKS," vol.11, 2015.

[14] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," 1982.

[15] J. C. Magee and D. Johnston, ". A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons.," 1997.

[16] S. T. Johnston, M. Shtrahman, S. Parylak, J. T. Gonc¸alves and F. H. Gage, "Paradox of pattern separation and adult neurogenesis: a dual role for new neurons balancing memory resolution and robustness.," pp. 60-68, 2016.

[17] R. C. O'Reilly and J. L. McClelland, "Hippocampal conjunctive encoding, storage, and recall: avoiding a trade-off. Hippocampus," pp. 661-682, 1994.

[18] M. W. Howard and M. J. Kahana, "A distributed representation of temporal context," 2002.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Peitro Perona, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollar, "Microsoft COCO: Common Objects in Context," 2015.

[20] Jozef Zurada. End effector target position learning using feedforward with error backpropagation and recurrent neural networks. In Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on, volume 4, pages 2633–2638. IEEE, 1994.

[21] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. Scientific data, 3, 2016.

[22] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 11. Z. Lipton, D. Kale, C. Elkan, and R. Wetzel. Learning to Diagnose with LSTM Recurrent Neural Networks. In International Conference on Learning Representations (ICLR 2016), 2016.

[23] Chao Ma, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering with memory-augmented networks. arXiv preprint arXiv:1707.04968, 2017.

[24] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1903–1911. ACM, 2017.

[25] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: A Convolutional Net for Medical Records. Journal of Biomedical and Health Informatics, 21(1), 2017.

[26] Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. Predicting healthcare trajectories from medical records: A deep learning approach. Journal of Biomedical Informatics, 69:218–229, May 2017.

[27] Aaditya Prakash, Siyuan Zhao, Sadid A Hasan, Vivek V Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Condensed memory networks for clinical diagnostic inferencing. In AAAI, pages 3274–3280, 2017.

[28] P. Kishore, S. Roukos, T. Ward and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," 2002.