Modeling the Rehearsal Effects of Humans

- Agarwal Mohit and Stevens Chrsitopher

Abstract

Forgetting, the inability to recall a specific piece of knowledge seems to be related to how humans learn and encode information into a neural network. As humans, we require numerous repetitions over time to be able to recall a specific piece of knowledge. Our project explores this notion of forgetting within the realm of artificial neural networks. In particular, we propose that forgetting is not the loss of information but rather the loss of fidelity as the neural network is shifted (i.e. the weights of each neural connection) towards the new concept. Thus forgetting is merely the transformation of pre-existing knowledge away from critical threshold weights stored within the neural network. The goal of the project is to model the learning process and demonstrate how an artificial neural network recalls learned concepts. Specifically we want to show the effect of how learning a new concept affects the recall of previously learned concepts.

We accomplished our goal by building a recurrent neural network which is capable of associating a vector space (concept) with the spelling of a word. The network is trained such that the original word is transformed into a vector space of related concepts (through the use of Google's word2vec algorithm), this vector space is used to train our RNN such that the spelling of the original word can be output through each iteration of the RNN.

Our preliminary results show that as new concepts are learned, the ability to recall old concepts degrade i.e. are forgotten. However, the ability to recall old concepts never fully degrade to zero but rather can only be recalled some fraction of the time, generally less than 50% of the time for all trials.

1. Introduction

Memory in cognitive science is an elusive subject. There has been much debate on how information is stored within the human brain, and conversely how do humans forget. We explored this idea of forgetting in an attempt to understand if there is a specific aspect of forgetting or if this is a byproduct of how the human neural net operates.

A central question about human cognition as related to computational cognition is how human cognition stores and recalls information. Related to this topic is how humans remember and specifically why we forget information. Some of the first works in this area were published by Ebbinghaus and his famous "forgetting curve." [1]. The results of his work, which were replicated by Murre, Japp, and Dros [2], shows that humans have a specific pattern of forgetting. In other words, humans tend to have fairly consistent patterns of learning, and subsequently forgetting. It was this study that became of interest when exploring the idea of forgetting.

To answer why or what causes the phenomena of forgetting, we began with the conjecture of storage limitations (maximum capacity). Our initial premise was that the human brain has limited storage capacity and forgetting is a result of this phenomena i.e. humans forget as a result of the reuse of neural connections, and biologically creating new neurons was expensive. We examined literature to understand if there was a real capacity limit of even an artificial neural network. Mejias and Torres work on capacity limits do show the complexity involved in estimating, and their work demonstrates that such a limit is not fixed but rather the result of some function which entirely depends on the training data and whether maximums and minimums can be found [3]. Additionally, even Watry's work [4] shows that human brain capacity limit is in the petabyte range, if not more, which was too much for us to reason about even within a simple model.

On a final note, regarding neural network capacity, we delved into basic mathematics in order to understand if capacity was indeed a limiting factor. It was here that we learned the basic premise was flawed. Coleen's work [5] on basic neural net mathematics shows that a given neuron operates by producing an output based on a weighted threshold function. This function can be expressed as a kernel density function, which probabilistically determines what the value of the output is based on some distribution of input values. This ties back into Mejias and Torres findings on a neural network capacity limit--data is not stored discretely but rather as a function of probabilistic distributions. Therefore estimating the capacity of even a simple neural network, artificial or biological, is difficult

since the density function is related towards the types of inputs. Another way of view this: a KDF can encode large quantities of data, especially if the "packing" of information is related.

Our interest in the forgetting curve, though was not displaced. The premise of why neural networks forget was still rather interesting and we wanted to build a model that could explore the notion of learning and forgetting. We decided to build a model that could explore the aspect of learning and from that model understand, at least from an artificial computational model, what is forgetting or "Do artificial neural networks 'forget' in similar patterns to the way humans forget?" It was this question that became our hypothesis for our research. Anecdotally, we wanted to see if we could model forgetting based on an artificial neural network.

In order to explore this aspect we again studied how humans learn information. In particular, we examined both short and long term memory recall processes in the attempt to build a simple model regarding learning and recall. Much of our model was based on Valiant's research and her work on "Memorization and Association on a Realistic Neural Model" [6]. The model we built was based on a recurrent neural network that could "spell" out a word based on learned associations. This model was the best exemplar for memory as it allowed us to explore the idea of learning concepts in a vector space rather than just using a computational mapping of word-to-spelling.

2. Memory Models Primer

Atkinson-Shiffrin were among the first to propose the computational model of memory in 1968 [7]. Their model was known as the multi-store model (Fig. 1), where information was input in the model through 'sensory register' to the 'short-term memory'. The information rehearsed inside short-term memory is transferred to 'long-term memory'. Baddeley and Hitch [8] proposed a more accurate model of working memory in 1974, which was further improved by Baddeley in 2000 [9]. The final model is a master-slave based system consisting of four significant entities namely 'Central Executive', 'Phonological Loop', 'Visuospatial Sketchpad' and 'Episodic Buffer' as shown in Fig. 2. Here 'Central Executive' is the master, which binds and organizes information in the coherent episodic manner. 'Visuospatial Sketchpad' and 'Phonological Loop' are associated with visual-spatial and verbal information respectively. The final component 'Episodic Buffer' links information across the above domains.

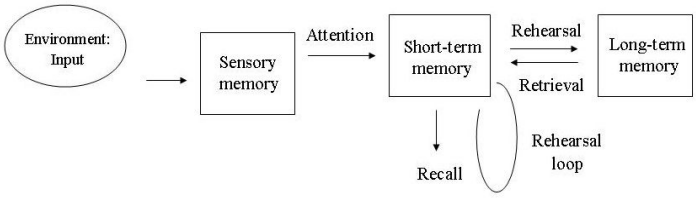

Figure 1: Atkinson Multi-Store model of memory

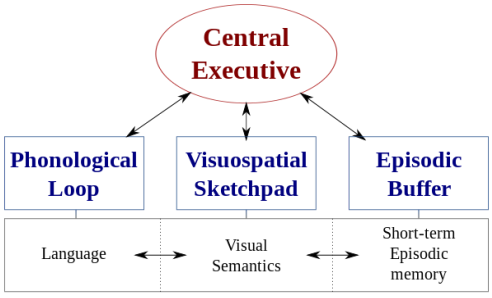

Figure 2: Baddeley & Hitch working memory model

3. Model

Here we explain our computational model which is based on neural networks. Before delving deep into the specific model we present a primer on neural networks which also provides the motivation behind the selection of our specific model.

3.1 Neural Network Primer

Neural networks are a computational approach to solve various machine learning problems (ex. Image recognition, weather prediction). The Artificial Neural Networks (ANNs) [10] includes inter-connected neurons, very roughly modeling the way a brain works (neurons with axon connections). Humans in, general, learn the new information based on the association with previously learned information. However, ANNs and Convolutional-Neural Networks (CNNs) [11] are incapable of modeling the stateless behavior pertaining to their model design. Recurrent Neural Networks (RNNs) [12] has been used widely to address this issue and persist previous information. They have been used in sequence modeling applications such as Natural Language Processing. However, traditional RNNs face problems of long-term dependencies of the information, because of imploding and exploding of gradients while

learning [13]. Long-Short Term Memory (LSTMs) does not face such problem as their design includes specific gates to throw out the irrelevant information [14]. In our model, we use the LSTMs to have the recurrent connections avoiding the gradient exploding problems.

3.2 The Neural Network Model

We build an LSTM based model of human memory to investigate the learning and forgetting effects of recall. The goal of the model is to learn the association of a word with its feature vector in vector space *v*. The model takes input a feature vector of the word, and tries to predict the spelling of the word. The model accomplishes the recall (i.e. spelling) by inputting the initial vector space, *v*, and the previous output into each subsequent iteration in a recurrent fashion.

The model denotes the network state (S), as a summation of the individual neuron outputs as following:

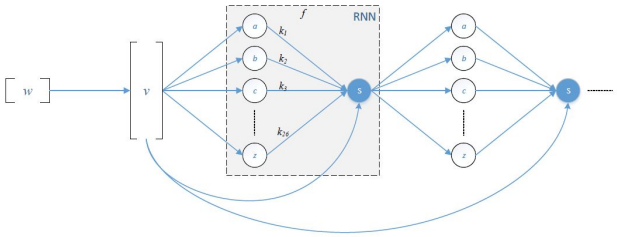

Figure 3: Exploded view of RNN model

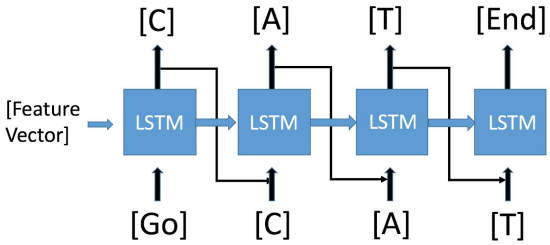

Figure 4: Neural Network implementation of the model

$$S = \bigcup_{i=1}^{|w|} f(i,v)$$

$$f(i,v) = v \bullet \sum_{j=0}^{26} k_j N_j \rightarrow w_i$$

(Fig. 3) S is dependent on the neural network threshold function *f,* which is expressed as a summation of the individual neuron states *N*. The variable *k* denotes the individual weights for that neuron. This function can be implicitly mapped back to the individual letter w_i, which allows the model to recall the spelling of the original word *w*.

Dataset and Preparation: We build the dataset of the model by scraping a text document, and learn associations of the words contained in the document using a skip-gram model of word2vec [15,16]. This maps each word in a vector space of size 16 (which we call feature vector of the word), where the mapping is related to the usage and concept of the words. For e.g. in the vector space, king - queen + male will roughly correspond to female. After preparing this large dataset, we pick the words of length 8 (to make the model simplistic) and their associated feature vectors as our dataset. The total number of words in our dataset was 7000 words. Here, in neural-network terminology, we treat 'feature vectors' as our 'training data' and 'word spelling' as corresponding 'labels'. With the above dataset in hand, we design our neural network model to predict word spelling from its associated feature vector. For e.g. [0.86 0.22 0.35 … 0.16] -> 'C A T'.

3.3 Implementation of the Neural Network Model

We implemented the above explained model in tensorflow[17]. Fig 4 describes the graphical representation of our model . As our feature vector is of size 16, we use 16 LSTM units in our model, and provide feature vector as the initial state to the model. The output of the LSTM at one time unit is fed as input for the next time step, accomplishing the recall (i.e. spelling). Word embeddings are used to transform the characters (ex. 'C', 'A' etc.) to the model state size. Gradient Descent algorithm is used with a learning rate of 0.001 to train the model.

4. Experiments and Results

Since the neural-network based models are probabilistic (not definite) we can not expect to see the exact same result with each re-run of the experiment. Hence, we repeat all of our experiments 100 times, and report the mean of the result of individual runs for each experiment.

4.1 Experiment 1: Number of iterations required to learn words in serial fashion.
In this experiment, we train our model to recognize a single word to 100% accuracy and we record the number of iterations necessary to achieve this goal. We randomly pick a word from the dataset (call it word1), train network on it and count the number of iterations to reach 100% accuracy of the particular word (i.e. training accuracy in our case). With this model trained already on word1, we retrain the model with word2 (also randomly picked) and note the required iterations to reach 100% accuracy. Such values are populated for in total of 5 words in Table 1.

	Word1	Word2	Word3	Word5
Mean	790.2882883	251.8198198	192.5405405	156.2792793
Std	301.2726759	123.3343717	105.8707431	65.43062546

Table 1: Required number of iterations to train words in a serial fashion

Here we see that it requires the higher number of iterations to learn the first word (~790) as compared to the iterations required to learn more words in a serial fashion (i.e. ~250 for word2, ~200 for word3).

4.2 Experiment 2: Accuracy of i words, after training i-th word with fixed iterations
Here, we explore the effect of learning new words in a serial fashion in terms of accuracy of the information learned so far. We randomly pick a word and train it for 2000 iterations, and report the accuracy of the words learned so far, which is word1. Further, we randomly pick a second word, train it for 2000 iterations, and report the accuracy of words learned so far i.e. word1 and word2. We keep repeating step in such fashion to populate the Table 2. Please note that specific accuracy of word_i will be close to 100% after the moment word_i is trained with ~ 2k iterations. We did the experiment with keeping the number of iterations as a parameter and found that results are consistent for 1k, 2k and 5k iterations.

	Word1	Word2	Word5	Word10	Word20	Word50	Word100
Accuracy	1	0.5980	0.3437	0.2462	0.1803	0.1389	0.1198

Table 2: Mean accuracy of i words after training word_i in a serial fashion with fixed iterations

After the training of the second word, the accuracy of both the words (word1 and word2) is ~60%, which further drops to ~25% (accuracy of 10 words) after training 10th word.

4.3 Experiment 3: The learning behavior for a particular word
In this experiment, we train the network with a particular word (randomly picked) and plot the training accuracy with each iteration in Fig.5.

4.4 Experiment 4: Forgetting behavior
In this experiment, we select a word randomly (word1) and train the model on it for 2000 iterations. For further 1500 iterations, we select different random words for training and evaluate the retention accuracy of word1 at each iteration. We keep repeating this behavior to see the forgetting and rehearsal behavior of a particular word (as shown in Fig. 6).

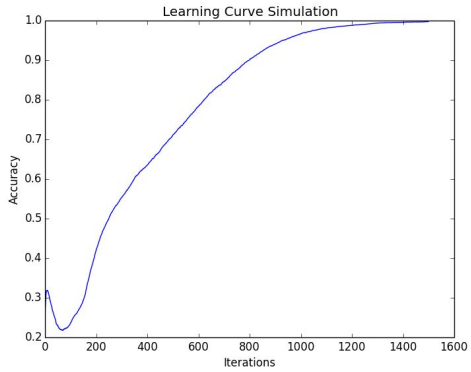

Figure 5. Learning curve

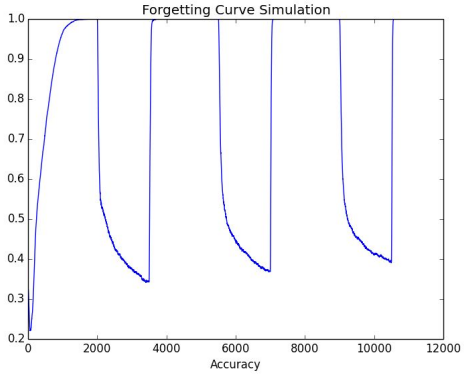

Figure 6. Forgetting curve

5. Results & Discussion

Our experiments do show that neural networks are capable of forgetting. Whether this result is similar to humans still remains an open question. However, our data does show that at least our model, is capable of learning similar to the way humans learn. Experiment 3 shows the typical s-curve that is associated with human learning, as discovered by Ebbinghaus in 1885. The s-curve as shown is a smoothed result from multiple trials of the same experiment, but just like humans, individual trials exhibit sharp increase and decrease in the accuracy. Furthermore, the data collected in both experiments 2 and 4 show that neural networks do "forget", but this "forgetting" is a product of shifting weights within the neural network and that new training data causes the network to "lose fidelity."

It is this last finding that is most interesting: neural networks do not forget information per se, but rather they lose fidelity due to shifting weights and thresholds. What happened within our experiments was that the neural networks could recall with near perfection until new training data was presented and this phenomenon was replicated with each subsequent input. We explored this idea of forgetting by retraining the neural network on a previously learned data and experiment 4 corroborated our hypothesis--the network did not forget, but rather its fidelity was altered and retraining only required a very small subset of repetitions to correctly retrain it to near perfect accuracy.

Based on the data collected, though still immature, indicate that artificial neural networks do experience a phenomenon similar to humans forgetting. This correlation represents interesting questions regarding learning and plasticity of neural networks--do humans forget because the network is always in a constant state of transformation due to continuous input of new data? Obviously our results are inconclusive and too naive to draw such a broad generalization but the correlation is too similar to ignore.

6. Future work

The notion of forgetting is still not well understood and though our experiments do show a tendency towards "forgetfulness" it is anecdotal at best. Many more experiments are necessary to fully understand the effect of fidelity loss with regard to new training data and further experimentation would necessary beyond that to explore the correlations with human learning and consequently forgetting.

One experiment that we did not accomplish was to train the neural network with multiple concepts together, in an alternating sequence, so as to allow the network to learn both concepts at the same time. This would be interesting to see the impact on the number of iterations at whether it impacts learning at all, In addition to this, it would also be interesting to see if there is some "maximum" training that allows the neural network to always have a near perfect recall.

One conjecture that has arisen as a result of our research and experimentation is the question regarding short-term memory: If neural networks "forget" based on shifting weights due to new input data, then is short-term memory just very specialized network that is so "plastic" that it shifts weights rapidly and cannot store information for long periods of time?

References

1. "Classics in the History of Psychology." *Classics in the History of Psychology -- Ebbinghaus (1885/1913) Chapter 1*. N.p., n.d. Web. 05 Dec. 2016.
2. Murre, Jaap M. J., and Joeri Dros. "Replication and Analysis of Ebbinghausâ Forgetting Curve." *PLOS ONE PLoS ONE* 10.7 (2015): n. pag. Web.
3. Mejias, Jorge F., and JoaquÃn J. Torres. "Maximum Memory Capacity on Neural Networks with Short-Term Synaptic Depression and Facilitation." *Neural Computation* 21.3 (2009): 851-71. Web.
4. Watry, G. “Brain's memory capacity is 10 times greater than previously thought.” *R & D*, Web. 21 Jan. 2016.
5. A. C. C. Coolen (4). "A Beginner's Guide to the Mathematics of Neural Networks." *Springer*. N.p., n.d. Web. 05 Dec. 2016.
6. Valiant, Leslie G. "Memorization and Association on a Realistic Neural Model." *Neural Computation* 17.3 (2005): 527-55. Web.
7. Atkinson, R.C.; Shiffrin, R.M. (1968). "Chapter: Human memory: A proposed system and its control processes". In Spence, K.W.; Spence, J.T. *The psychology of learning and motivation (Volume 2)*. New York: Academic Press. pp. 89–195
8. Baddeley, A.D., & Hitch, G. (1974). *Working memory.* In G.H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 8, pp. 47–89). New York: Academic Press.
9. Baddeley, A.D. (2000). "The episodic buffer: a new component of working memory?". *Trends in Cognitive Science*. **4**: 417–423
10. Artificial Neural Networks: https://en.wikipedia.org/wiki/Artificial_neural_network
11. Convolutional Neural Networks: https://en.wikipedia.org/wiki/Convolutional_neural_network
12. Recurrent Neural Networks: https://en.wikipedia.org/wiki/Recurrent_neural_network
13. The vanishing gradient problem: http://neuralnetworksanddeeplearning.com/chap5.html
14. Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory" (PDF). *Neural Computation*. **9** (8): 1735–1780
15. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).
16. Word2vec: https://www.tensorflow.org/versions/r0.12/tutorials/word2vec/index.html
17. Tensorflow: https://www.tensorflow.org/
18. Kahnt, Thorsten, Soyoung Q. Park, John-Dylan Haynes, and Philippe N. Tobler. "Disentangling Neural Representations of Value and Salience in the Human Brain." *Proceedings of the National Academy of Sciences* 111.13 (2014): 5000-005. Web.