Fast Weights Using Improved Memory Consolidation Designs

Bowen Xu[1], Jimmy Ba[2], Richard Zemel[1]

Department of Computer Science, University of Toronto[1]

Department of Electrical & Computer Engineering, University of Toronto[2]

Abstract

Storing better quality and greater amount of memory has been a difficult challenge for deep learning. While the current artificial neural networks excel at tasks such as regression and classification, they fail to perform equally well on representing variables and storing data over long time. We introduce an enhanced fast weights model that includes robust memory consolidation designs into the existing memory transformation mechanisms. We show that our model comes with no additional costs, converges faster, and out-performs the original fast weights model on the associative retrieval tasks.

Keywords: Fast Weights; Memory Consolidation; Memory Transformation; Deep Learning; Associative Retrieval

Background

Recently, Benna and Fusi have demonstrated that the combination of a power-law memory decay with a fast new memory adaptability maximizes both the memory storage and lifetime (2016). Their method treats each piece of memory as random and uncorrelated, and maximizes the signal to noise ratio (SNR) of all the stored memories (Benna & Fusi, 2016). Since Benna and Fusi use a power-law decay function which has the form of $t^{-\gamma}$, $\gamma > 0.5$ is required for the function to converge (2016). In addition, Benna and Fusi also suggest fast adaptability for the memory system which implies that the learning rate for new information should be exactly 1.0 (2016).

$$w_a(t) \equiv \sum_{t'<t} \Delta w_a(t') r(t-t') \tag{1}$$

$$S_{t'}(t) \equiv \frac{1}{N} \left\langle \sum_{a=1}^{N} w_a(t) \Delta w_a(t') \right\rangle \tag{2}$$

$$N^2_{t'}(t) \equiv \left\langle \frac{1}{N^2} \left(\sum_{a=1}^{N} w_a(t) \Delta w_a(t') \right)^2 \right\rangle - S^2_{t'}(t) \tag{3}$$

$$S/N(t-t') = \sqrt{\frac{N r^2(t-t')}{\sum_{t"<t, t" \neq t'} r^2(t-t")}} \tag{4}$$

$$\sum_{t"<t, t" \neq t'} r^2(t-t") \approx \int_1^{\infty} r^2(t) dt \tag{5}$$

Recent advances in deep learning (Graves et al., 2016; Ba, Hinton, Mnih, Leibo, & Ionescu, 2016) have also shown that the use of an external memory with the artificial neural networks can significantly improve their abilities to retain information. Ba et. al. devise a bidirectional memory transformation system. Knowledge is not only transferred from the fast weights to the memory cells of a Recurrent Neural Network (RNN) through its hidden vector update but also transitioned from the memory cells to the fast weights as it is updated using the latest hidden vector at each time step. Also, a learning rate η and a decay rate λ are included to control how much the fast weights should store new knowledge versus how quickly it should forget previous knowledge.

$$A(t) = \lambda A(t-1) + \eta h(t) h(t)^T \tag{6}$$

$$h_s(t+1) = f(LN(Wh(t) + Cx(t) + A(t) h_{s-1}(t+1))) \tag{7}$$

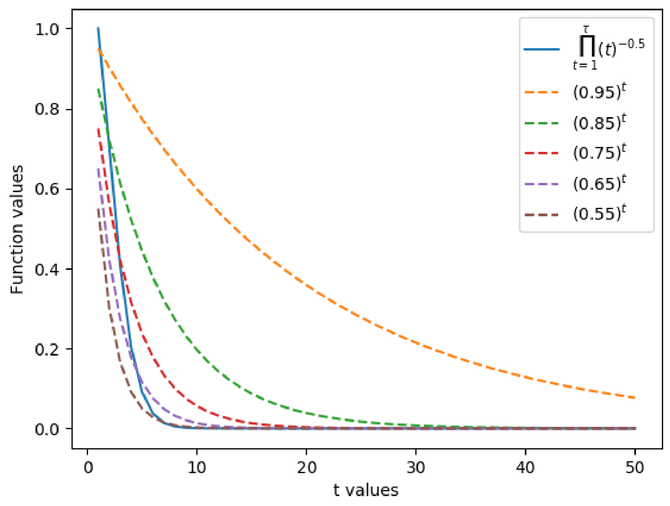
$$h(t+1) = h_s(t+1) \tag{8}$$

Our Contribution

We incorporate memory consolidation designs from (Benna & Fusi, 2016) to the fast weights model (Ba et al., 2016). We keep the fast weights setup and the weights transferring mechanism unchanged. However, our fast weights uses a learning rate of $\eta = 1.0$ instead of $\eta = 0.5$. We also substitute the original exponential decay function $\lambda = 0.95^t$ to a power-law decay function $\prod_{t=1}^{t=\tau} t^{-0.5}$ where τ is the number of time steps elapsed since the initial storage of a specific memory.

Decay Functions Comparison

Figure below shows that our power-law decay function decays much faster than the original exponential decay function. We believe that because new information is stored in a greater portion than before, the fast weights has to decay its existing memory quicker to prevent memory interference. This design also forces the RNN to learn patterns quicker from the fast weights since existing information is quickly removed.



Experiments and Results

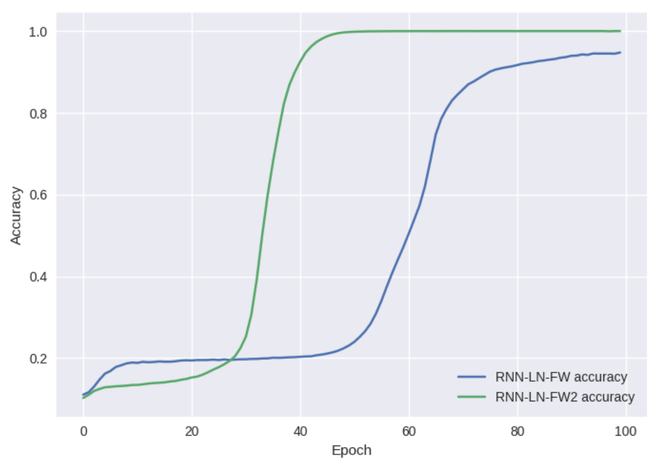
We evaluate three RNN models on the associative retrieval tasks on three difficulty levels. Our data consists of sequences of key-value pairs concatenated with two question marks and one query key. The keys are lower case characters from the English alphabet chosen randomly without replacement, and the values are digits among 0 to 9 chosen randomly with replacement.

Input String	Target
c9k8j3f1??c	9
j0a5s5z2??a	5

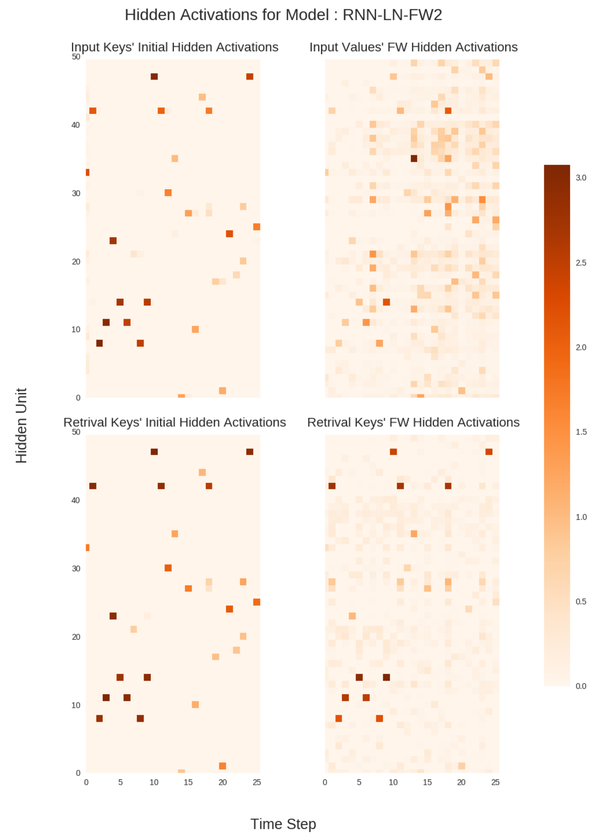
Our training data contains $100,000$ sequences and our testing data contains $50,000$ sequences. The difficulty levels are three key-value pairs (K $= 3$), four key-value pairs (K $= 4$), and twenty-six key-value pairs (K $= 26$). The following models are evaluated on correctly predicting the associated value of the query key: a vanilla RNN (CONTROL), the original fast weights model (RNN-LN-FW), and the improved fast weights model (RNN-LN-FW2). All RNN models contain a single hidden layer with 50 hidden units and are trained with 100 iterations.

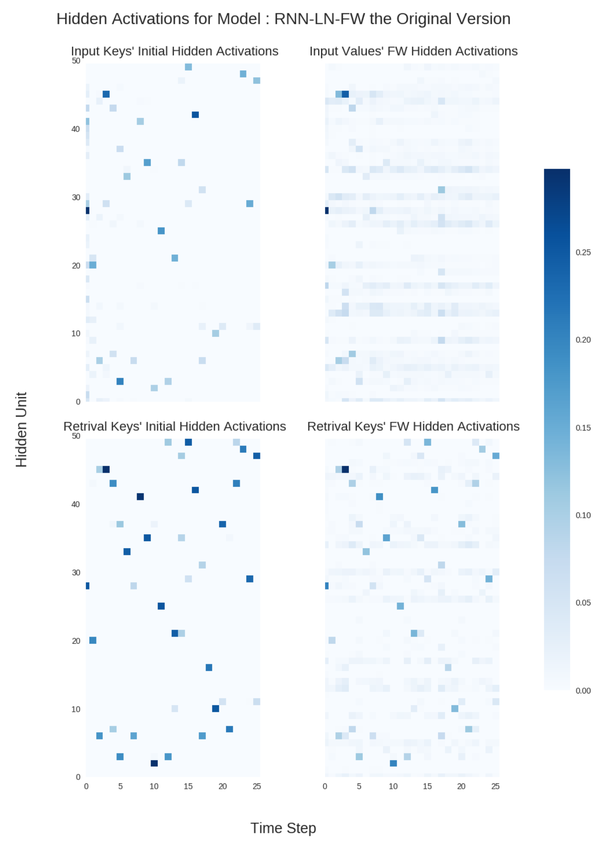
Model	K $= 3$	K $= 4$	K $= 26$
RNN-LN-FW2	$\mathbf{99.74\%}$	$\mathbf{100\%}$	$\mathbf{100\%}$
RNN-LN-FW	99.5%	99.71%	94.75%
CONTROL	42.71%	34.49%	12.07%

FIGURE 1: Testing Accuracy K=26



Hidden Activations Visualization





Conclusion and Future Work

We introduce an enhanced fast weights model by incorporating memory consolidation designs from (Benna & Fusi, 2016). We show that our model works better on the associative retrieval tasks and converges faster. Our next direction is to evaluate the models on more difficult tasks such as sorting and repeated-copying. We also intend to investigate other decay functions for the fast weights model.

References

Ba, J., Hinton, G., Mnih, V., Leibo, J. Z., & Ionescu, C. (2016, October). Using Fast Weights to Attend to the Recent Past. *ArXiv e-prints*.

Benna, M., & Fusi, S. (2016). Computational principles of synaptic memory consolidation. *Nature Neuroscience*, *19*, 1697–1706.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwinska, A., ... Hassabis, D. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471–476.