Extending Memory Capacity of Neural Associative Memory based on Recursive Synaptic Bit Reuse

Tianchan Guan[*,+], Xiaoyang Zeng[*], Mingoo Seok[+]
[*]Fudan University, Shanghai, China, [+]Columbia University, New York, NY USA

***Abstract*—Neural associative memory (AM) is one of the critical building blocks for cognitive workloads such as classification and recognition. It learns and retrieves memories as humans brain does, i.e., changing the strengths of plastic synapses (weights) based on inputs and retrieving information by information itself. One of the key challenges in designing AM is to extend memory capacity (i.e., memories that a neural AM can learn) while minimizing power and hardware overhead. However, prior arts show that memory capacity scales slowly, often logarithmically or in squire root with the total bits of synaptic weights. This makes it prohibitive in hardware and power to achieve large memory capacity for practical applications. In this paper, we propose a synaptic model called recursive synaptic bit reuse, which enables near-linear scaling of memory capacity with total synaptic bits. Also, our model can handle input data that are correlated, more robustly than the conventional model. We experiment our proposed model in Hopfield Neural Networks (HNN) which contains the total synaptic bits of 5kB to 327kB and find that our model can increase the memory capacity as large as 30X over conventional models. We also study hardware cost via VLSI implementation of HNNs in a 65nm CMOS, confirming that our proposed model can achieve up to 10X area savings at the same capacity over conventional synaptic model.**

Keywords—Associative Memory; Synaptic Model; Memory Capacity; Hopfield Neural Network

I. INTRODUCTION (*HEADING 1*)

Associative memory (AM) is a type of memory that can learn memories and the memories are retrieved by the learned memories. This makes AM distinct from traditional computer memory that retrieves data by the address of the data. A well-known type of AM is content addressable memory (CAM) which retrieves memories through brute-force comparison between input and stored data. On the other hand, another AM, called neural AM, learns memories just like humans brain does by updating plastic synaptic strengths (weights). A well-known example of neural AM is Hopfield neural network (HNN). AMs can be used for traditional systems such as databases and routers [1,2] and also emerging cognitive systems which perform classification, recognition, and clustering [3,4].

One of the key objectives in designing neural AM is to enable large memory capacity, i.e., learn many memories, while consuming low power and compact hardware area. To achieve this, it is critical to create an efficient synaptic model since synapses not only determine memory capacity but also take a dominant amount of area and power dissipation. For example, in recurrent neural AM designs, synapses can outnumber neurons by a factor of 1000X.

However, conventional synaptic models are not very efficient to increase memory capacity. In the HNN, one way is to increase the bit-width of synaptic weights. This can increase memory capacity in the beginning but cannot increase any more after a certain point (see Fig. 4). This is because only a fractional portion of the added bits are utilized in the learning process. Another way is to increase the number of synapses. For fully connected networks like HNN, this means we need to increase the number of neurons. However, prior studies theoretically show that the memory capacity scales only logarithmically or in squire root with the number of synapses [5]. This makes neural AM with the conventional synaptic model prohibitive in area and power to achieve a large memory capacity for practical applications.

In this paper, we aim to devise a synaptic model that can better utilize available synaptic bits in increasing memory capacity. We propose a new model titled *recursive synaptic bit reuse*. The main idea is that AM requires different synaptic bit-widths for learning and retrieving memories: it requires wider synaptic bit-width for learning but it works reasonably well with narrower synaptic bit-width for retrieving. Therefore, after learning a set of inputs, we can fix a few MSBs for retrieving. Then in the next learning phase we can reuse the remaining synaptic bits to learn another set of inputs. We can continue this process recursively until we freeze all the synaptic bits. We find this model can increase memory capacity with synaptic bit-widths, near-linearly and better than prior models [6,7]. We do not need to add more neurons and synapses, either.

In addition, we find our model can provide extra robustness in learning inputs that are correlated. The HNN is known to lose memory capacity for correlated inputs since they confuse networks. Our model handles inputs across multiple learning phases. If correlated inputs enter in different learning phases, they would not interfere, and thus enable larger memory capacity.

We experiment our proposed synaptic model in HNNs containing synaptic bits ranging from 5kB to 327kB, using both fabricated-random and real image data. We also study VLSI hardware implementation of those networks. The experiment results show that our proposed model can achieve 30X greater memory capacity than the conventional model for the same amount of total synaptic bits. At the same memory capacity, our proposed model can save >10X silicon area to implement a HNN. We anticipate that the savings would be more significant if we consider a larger network.

II. HOPEFIEL NEURAL NETWORK

HNN is one of the most influential and widely used neural AMs, proposed by Hopfield [8] in 1986. It is a symmetric recurrent neural AM with fully connected synapses. The state of each neuron is +1 or -1. And the network updates the synaptic weights using hebbian rule to memorize new inputs.

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

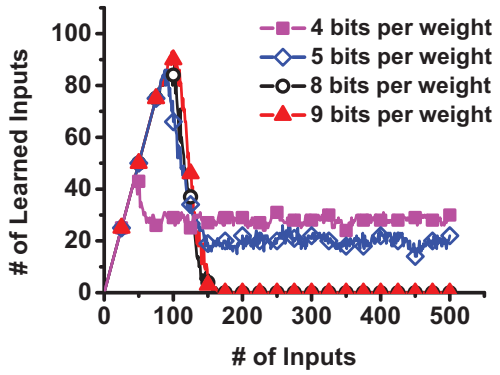

Fig. 1.Learning dynamics of 500-neuron HNN

A. Causes of Limited Memory Capacity

We investigate the causes that limit memory capacity using a 500-neuron HNN. As shown in Fig. 1, we simulate the number of inputs that the network learns across different synaptic bit-widths from 4 to 9 bits per synaptic weight. Fig. 1 shows that generally, wider bit-widths enable larger memory capacity. One notable feature is the large drop in capacity after a certain number (~100) of inputs are learned. The capacities drop to zero for the wider bit-width cases (8 and 9 bits per weight). This dynamics is called *blackout catastrophe* [10] and caused by interference among inputs. Every time AM learns a new input it changes the same set of synapses, and thus perturbing previously-learnt memories. With wider synaptic weights, it incurs a larger amount of interference and eventually makes all the memories irretrievable. A common solution is to avoid HNN to learn additional inputs as the HNN reaches the peak capacity point.

Other than the blackout catastrophe, the causes to limit memory capacities are:

1) *Limited synapse bit-width*. Too narrow bit-width limits the memory capacity. However, once we have a sufficient bit-width this bound issue become negligible.
2) *Limited number of synapses*. Having more number of synapses can increase memory capacity. However, the memory capacity scaling is not very favorable.
3) *Inputs correlation*. The capacity depends on the characteristics of inputs that HNN learns. If inputs are highly correlated with one another, HNN has a hard time to learn them, having limited memory capacity.

III. RECURSIVE SYNAPTIC BIT REUSE MODEL

A. Intuition behind our recursive synaptic bit reuse model

We have investigated to build a new synaptic model that has a better efficiency to increase memory capacity than increasing bit-with or the number of synapses in the network. Our proposed model is based on the fact that the required bit-widths for learning and retrieving are quite different. Prior studies have shown that narrow synaptic weights can retrieve data well as long as learning is done properly [5]. Fig. 2 shows the results of our experiment of 500-neuron HNN. The memory capacity degrades moderately as synapses lose more LSBs. When a single bit is used for each synaptic weight, we find that memory capacity decreases only by 17.9% in average. This implies that we can achieve roughly 160% of the conventional memory capacity by repeating the learning and clipping process only two times. The capacity will increase further as we repeat the process across all seven bits. Therefore, we can have HNN to learn a set of inputs with wide bit-width weights and fix only a few MSBs for retrieving. Then using the remaining bits, we can have HNN to learn another set of inputs.

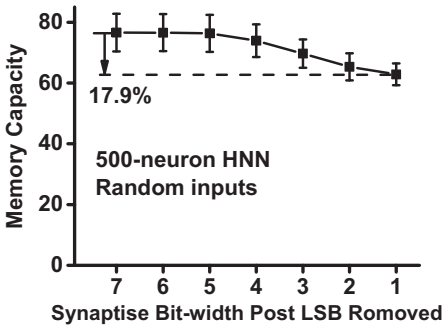

Fig. 2. Memory capacity of clipped synapse

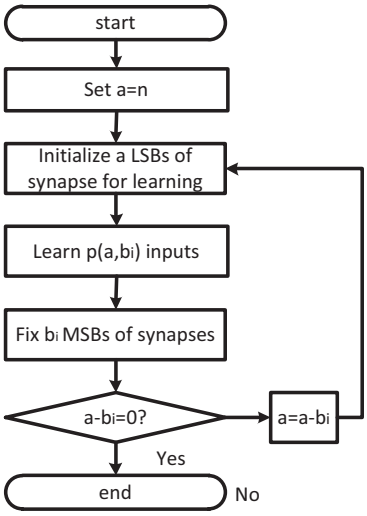

Fig. 3. Learning process of the proposed model

B. Proposed Synaptic Model: Details

Fig. 3 shows the learning procedures of our proposed synaptic model. Considering that a HNN has n-bit synaptic weights, we start by setting the bit-width used for learning in this learning phase (a) to n. We then initialize and start to train the HNN. We have a threshold parameter, $p(a,b_i)$, which is defined as the number of memories that can be retrieved by the b_i MSBs of a-bit synapses. This parameter can be pre-defined based on the characteristics of inputs that the HNN learns. The HNN learns up to $p(a,b_i)$ inputs since the clipping process can discard the inputs after $p(a,b_i)$ inputs. Then we fix the b_i-bit MSBs of the a-bit synapses. If the synapses still have bits that are not fixed, we can perform another phase of learning by setting a to $a - b_i$. This procedure is repeated until all the synapse bits are fixed.

The HNN can retrieve data by performing the regular retrieval procedure using each set of b_i-bit synaptic weights. If the HNN cannot retrieve data with the first set, it continues

with the next set, until the data is successfully retrieved or all sets are used. In this paper, we mostly use 1 for b_i, i.e., only a MSB is fixed but b_i can be further optimized in each learning phase for different network sizes and synaptic bit-widths.

C. Memory Capacity Benchmark

We compare our synaptic model with the conventional model in memory capacity across a range of synaptic bit resources in the context of HNN. We sweep different sizes and synaptic bit-widths of HNN For the conventional model, we change either the number of synapses or the bit-width of synapses. For our proposed model, we use b_i=1.

Fig. 4 shows the results of the comparisons, confirming that our proposed synaptic model achieves a significantly larger amount of memory capacity than the conventional models for the same amount of bits used in synapses. For the fixed network size (i.e. the fixed number of neurons and synapses), we can increase capacity by increasing bit-widths but this approach provides only diminishing returns after a certain point (orange line with triangle symbols). After that, we need to increase the size of networks to increase the number of synapses (red lines with circle symbols). But the efficiency is not very high. You need to increase the total synaptic bits from 40,640 to 2,617,216 to increase the capacity from 19 to 129.

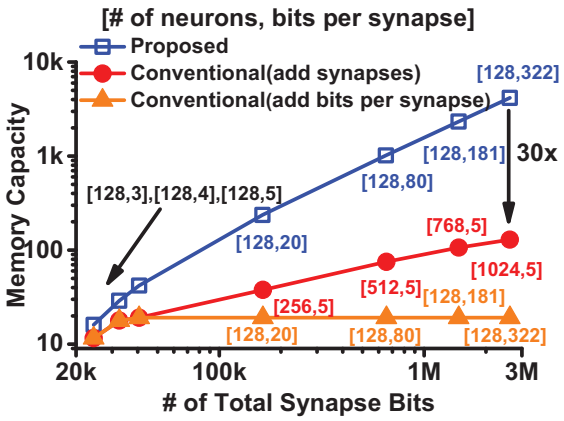

Fig. 4.Memory capacity comparisons

Also, our synaptic model enables HNN to increase memory capacity more efficiency without saturation. We do not need to enlarge a HNN because we can increase synaptic bit-widths for more learning phases. When the total synapse memory is ~327kB, our proposed synaptic model achieves 30X larger memory capacity than the conventional models. We expect a greater amount of improvement at larger synaptic memory.

D. Robustness to Correlation in Inputs

Our proposed synaptic model is expected to provide extra robustness to inputs having correlation, particularly if HNN learns those correlated inputs in different learning phases. This is because for proposed synaptic model the inputs stored in different bit positions of synapses have no interference with each other. Experiment shows that if the correlation level increases from 0 to 20%, the conventional model loses ~36.9% of memory capacity, while our model loses only 14.3%. When the correlation level increases further to 40%, our proposed model loses 36.9%, again about a half what the conventional model loses (65.5%).

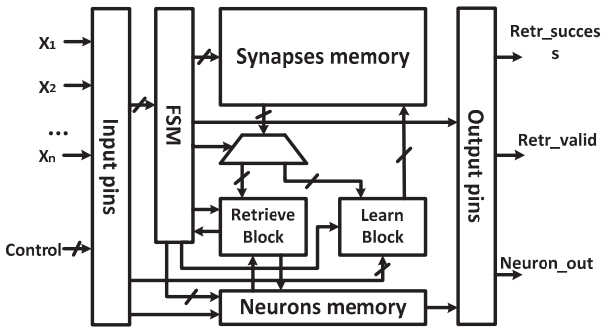

Fig. 5. Implementation of HNN with the proposed model

IV. HARDWARE EXPERIMENTS

A. Hardware Architecture

To evaluate the efficacy of our proposed synapse model, we design hardware architecture for the HNN (Fig. 5). It consists of a memory block for storing synaptic weights (Synapses memory), a memory for storing neuron's states (Neurons memory), a computing block for retrieving (Retrieve block), a computing block for learning (Learn block), and an FSM to generate address and other control signals.

The architecture performs learning by accessing the synapses memory. The FSM routes synaptic weights to the learn block one by one. The learn block then calculates each new weight based on the input. The FSM writes the updated weight in the synapses memory. The retrieval operation starts by writing input in the neuron memory. The retrieve block then updates each neuron state using the states of other neurons and the weights of the connected synapses. As described in Sec. II, we use the asynchronous retrieval method. Upon the end of retrieval, the FSM sets a signal called *retr_success* to 1 if the HNN successfully recalls the input. Also, the FSM outputs the retrieved data via a port called *neuron_out* sequentially while asserting a signal called *retr_valid* to indicate that the signal on the *neuron_out* is valid.

Algorithm 1 Learn Block
Input: *synp_in*[*n*-1:0]
synp_msk[*n*-1:0]
synp_msb[*n*-1:0]
Xin1
Xin2
Output: synp_out[n-1:0]
Function:
if *XOR(synp_in AND synp_msb)* == 1 **then**
synp_sign[*n*-1:0] = *n-bit AND* (~*synp_msk*)
else
synp_sign[*n*-1:0] = *n-bit* 0
end if
synp_fix[*n*-1:0] = *synp_in AND* (~*synp_msk*)
synp_pls[*n*-1:0] = *synp_sign XOR* (*synp_in AND synp_msk*)
synp_max[*n*-1:0] = *synp_msb XOR synp_msk*
synp_min[*n*-1:0] = *n-bit* 1 *XOR synp_msk XOR synp_msk*
if *synp_pls* == *synp_max OR synp_pls* == *synp_min* **then**
synp_update=synp_pls

else if *Xin1 == Xin2* **then**
synp_update=synp_pls+1
else
synp_update=synp_pls-1
end if
synp_out=synp_fix XOR (synp_update AND synp_msk)

We implement our proposed *synaptic recursive bit reuse model* in the learn block. Algorithm 1 shows the steps to perform the proposed model. With the help of control signal *synp_msb* and *synp_msk,* which are for indicating the position of MSB of synaptic weights in current phase and masking MSBs that are not for learning, the learn block can operate with operands with same bit-width in every phase, thus, the computation resource can be saved.

B. Experiment Results and Comparisons

We implement the HNN architecture in a 65nm high V_{TH}. We write RTL-netlists and synthesize them using an industrial standard cell library and Synopsis Design Complier. The memory is implemented in latch cells in the library.

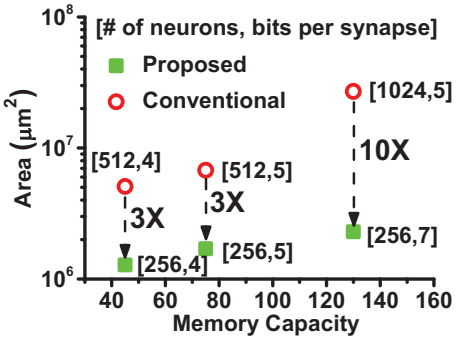

Fig. 6. Memory capacity and area of HNN VLSI Implementation

We compare the areas of the HNN hardware using the conventional and the proposed synapse models. We sweep the number of neurons and the bit-width of synaptic weights for both HNNs. Through the sweep we can find several designs that achieve target capacities of 45, 75 and 130. The HNN based on the conventional synaptic model cannot reach any of the target memory capacities with reasonably wide weight bit-widths. Therefore, we increase the number of neurons to 512 and 1024. On the other hand, the HNN based on the proposed model can achieve all the target capacities with 256 neurons only by increasing the bit-widths of synaptic weights.

Fig. 6 shows that the area of the HNNs that meet the target memory capacities. For the target capacities of 40 and 80, we find that the HNNs using our proposed model become 3X smaller than those using the conventional model. For the target capacity of 130, the HNN using the proposed model becomes 10X more area-efficient than the conventional HNN. We expect our proposed model becomes even more area-efficient for larger capacity targets. This is because the conventional model requires increasing the number of neurons and therefore synapses for having larger memory capacity. Our model only requires increasing synaptic bit-width. Also, it uses each bit more efficiently by taking advantage of the different bit-width requirements for learning and retrieving.

V. Conclusion

In this paper, we propose a new synapse model titled *recursive synaptic bit reuse* to scale memory capacity of neural AM in a highly area-efficient manner. The model leverages the different bit-width requirements of learning and retrieving in neural AM, enabling a neural AM can use each bit more efficiently. The efficacy of the proposed model is evaluated against the conventional model. The results show that moderate-size HNN based on our proposed synapse model can learn 30X more inputs than that based on the conventional models for the same amount of total synapse bits. The proposed model also enables HNN to learn inputs that are correlated, almost by 2X more robustly as compared to the conventional model. In our investigation on hardware cost, our proposed synapse model can achieve as high as 10X area reduction to achieve the same target memory capacity.

VI. Acknolwdgement

The project is supported by Research Initiatives for Science and Engineering (RISE) and China Scholarship Council. We appreciate Stefano Fusi (Columbia) for technical discussions.

References

[1] C. S. Lin, D. C. Smith, and J. M. Smith, "The design of a rotating associative memory for relational database applications," *ACM Transactions on Database Systems (TODS)*, vol. 1, no. 1, 1976.

[2] N.-F. Huang, W.-E. Chen, J.-Y. Luo, and J.-M. Chen, "Design of multi-field IPv6 packet classifiers using ternary cams," in *IEEE Global Telecommunications Conference*, 2001, pp. 1877–1881.

[3] B.-L. Zhang, H. Zhang, and S. S. Ge, "Face recognition by applying wavelet subband representation and kernel associative memory," *IEEE Transactions on Neural networks*, vol. 15, no. 1, pp. 166–177, 2004.

[4] A. V. Gavrilov and S. Lee, "An Architecture of Hybrid Neural Network Based Navigation System for Mobile Robot," *Seventh International Conference on Intelligent Systems Design and Applications* (ISDA 2007), Rio de Janeiro, 2007, pp. 587-590.

[5] M. K. Benna and S. Fusi, "Computational principles of biological memory," arXiv:1507.07580, 2015.

[6] A. Mehta and J. Luck, "Power-law forgetting in synapses with metaplas-ticity," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 09, 2011.

[7] M. C. Van Rossum, M. Shippi, and A. B. Barrett, "Soft-bound synaptic plasticity increases storage capacity," *PLoS Comput Biol*, vol. 8, no. 12, p. e1002836, 2012.

[8] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[9] K. F. Cheung, L. E. Atlas, and R. J. Marks, "Synchronous vs Asynchronous Behavior of Hopfield's CAM Neural Net," *Applied Optics*, vol. 26, no. 22, 1987.

[10] J. M. Brader, W. Senn, and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Computation*, vol. 19, no. 11, pp. 2881–2912, 2007.