Sparse Signal Reconstruction via Kanerva's Sparse Distributed Memory

Ruslan Vdovychenko
V. M. Glushkov Institute of
Cybernetics, NAS of Ukraine
Kyiv, Ukraine
ruslan.vdovichenko1@gmail.com

***Abstract*—The phenomenon of memory has been studied by many neurobiologists. The variety of memory types includes short-term, long-term, sensory, topographic, semantic, immune, and so on. However, none of them works as a memory of the electronic devices: new information does not overwrite the old one, losing a small number of neurons does not affect the whole system, there is no explicit separation into address and data. Human memory is also capable of recognizing complex objects and structures without great efforts. The purpose of this article is to observe the representation of high-level objects with a set of features and relations in an artificial intelligence system. The specific sparse feature-based encoding is introduced. This representation is investigated subject to fast GPU implementation of a modified version of a well-known human memory model called Sparse Distributed Memory (SDM). A hybrid model of SDM and Compressed Sensing is proposed. Two techniques for reading sparse data from the hybrid model are designed and examined. Comparative analysis for both sparse and dense signal reconstruction problems is provided.**

Keywords—GPU, SDM, sparsity, neural network

I. Introduction

A. Classic Model Overview

Sparse Distributed Memory is the classical human memory model designed by Pentti Kanerva [1]; one of its main theoretical advantages is the relation to cerebellar models and, in particular, models of Marr and Albus [1]. SDM is also related to immunological memory models [2].

Kanerva's SDM model consists of N M-dimensional integer vectors that serve as M-bit physical memory cells. It means that one bit of data doesn't correspond to one bit of memory, but an integer. L-bit binary addresses are used for addressing; that is, physical cells are much smaller than addresses (sparsity) [1]. Data are M-bit binary numbers. In the classical Kanerva's design, an L-bit binary address is associated with each physical cell.

Both reading and writing operations, unlike computer memory, involve several activated cells. In classic Kanerva's model, the cell is activated if the Hamming distance (i.e., the number of positions at which items differ) between the input address and the cell address does not exceed a predefined threshold d [1].

Meanwhile, Jaeckel's approach suggests associating each memory cell with a short K-dimensional mask ($K \ll L$). Every index of the mask is associated with a target value of a corresponding bit. The resulting model has numerous advantages: it is compatible with the Marr-Albus model of the cerebellum [3], its implementation is faster due to the reduction of the number of comparisons.

B. Dense Signal Reconstruction Experiment

Dense signal reconstruction via fast GPU-based SDM implementation has been studied recently [4]. Jeackel's modification was chosen for the layer's activation function [5]. Some of the insights provided by the experiment include:

1. Strong mask length dependency

Activation mask length is the most crucial parameter of the model. It affects read/write operations performance as well as data restoring accuracy. The longer the mask, the fewer data samples are matching the mask and, therefore, get ignored by the model. The longer the mask, the longer is activation function processing. Designing and investigating the algorithm that estimates the optimal mask length w.r.t the specified metrics is an open and challenging problem [6].

2. High model stability

SDM noise tolerance estimation showed that a satisfactory signal recovering level is achieved even for two noisy duplicates per data sample.

C. Sparse Signal Reconstruction Problem

However, the dense signal (e.g., a sequence of image pixels or a sentence encoded via Word2Vec or similar technique) does not possess a complex structure. Instead, suppose that we have to deal with some meaningful attributes (for example, a tree might be old or young, tall or low, with or without leaves, etc.). In this case, it turns out that a single object description consists of a small set of features, even though a global variety of observed features can be vast.

The described design of a problem leads to the necessity of a sparse and discrete data processing. More specifically, a neural network for storing sparse discrete or binary vectors is needed. Some of the additional requirements to the model's characteristics include but not limited to:

- distributed representation of data with some level of fault tolerance against errors occurring in separate areas of memory;
- information retrieval is associative, i.e., reading from memory works even for an incomplete set of object's features; this property refers to the human brain's ability to generalize input data that is based on incomplete or fuzzy patterns.

D. Previous Results on Sparse Encoding

Some results and propositions regarding sparse data representations for SDM were published by Sjodin [7]. He examined an encoding design called Sparchunk coding. Its main idea is restructuring input data into a recursive sequence via a specific noncommutative and nonassociative

binary operation called chunking. This approach has one critical drawback: its calculations are hard to parallelize. This disadvantage makes Sparchunk encoding unapplicable for real-world problems.

Another known technique successfully applied to SDM is the N-of-M code [6]. It deals with probabilistic error correction right after reading.

E. Compressed Sensing

Compressed Sensing is a signal processing technique that allows recovering signals from the observations with a sample rate that is much lower than The Nyquist–Shannon sampling theorem requires [8][9].

In Compressed Sensing schema, input signal X is projected onto the basis where it is sparse, and the projection operator satisfies Restricted Isometry Property (RIP) [10]. The signal is recovered by finding solutions to undetermined linear systems.

This technique has been successfully applied for numerous real-world problems, including MRI [11], facial recognition [12], space researches [13].

F. Proposed Algorithm

This paper suggests an integrated model constructed as a hybrid of SDM and Compressed Sensing. Its design is aimed at the efficient storing of high-level objects and sparse/discrete vectors. The model is highly scalable for GPU clusters.

II. Model Specification

A. Activation function

Since we deal with sparse signals, each data sample is a strongly imbalanced $0/1$ array, neither class Kanerva's activation nor Jaeckel's modification suit the experiment. The number of cells for SDM is $651 * (650 - 1)/2 = 211575$.

The activation function for the sparse signal reconstruction experiment was constructed the following way: each cell was associated with a pair of indices (i, j) without repetition. Therefore, a cell was activated for reading/writing only if a data sample possesses (i) and (j) features [7]. This approach deals with internal class imbalance definitive for each encoded sample.

B. Reading techniques and threshold selection

Suppose that an image possesses k features out of 651. Then the probability of 1 is $p_1 = k/651$, the probability of 0 is $p_0 = 1 - p_1$.

If i-th cell contains w records, then a single item contains $w * p_1$ ones on average. For this cell, the threshold is calculated in the following way:

$$threshold_i = w * (p_1 - p_0)$$

1) "Statistical" approach

This approach extends the basic SDM model with a threshold.

We calculate thresholds for each activated cell; then we get a general threshold in the following way:

$$threshold = Median(\{threshold_i\}_{i=1}^{n})$$

Then this threshold is used instead of 0 in classic SDM reading operation.

2) Parameterized "biological" approach

Suppose that there are C activated cells. The parameter $\alpha \in (0, 1)$ should be chosen beforehand. for each index j of each activated cell i we calculate an array of index wise decisions:

$$decisions_{i,j} = sign(cell_{i,j} - threshold_i)$$

Then, the result is calculated as:

$$result_j = \left(\sum_{i=1}^{C} decisions_{i,j}\right) \geq \alpha * C$$

This research deals with both "statistical" and "biological" models $(\alpha \in \{0.01,\ 0.1,\ 0.25,\ 0.5,\ 0.75,\ 0.9,\ 0.99\})$.

III. Description of the Experiment

A. Dataset

CIFAR-10 dataset, which is widely used to test, validate, and train machine learning and computer vision algorithms, was selected. The dataset consists of 60000 images. Images are divided into ten classes (birds, cats, cars, etc.), 6000 images per category.

The binary version of the dataset is the most convenient for the experiment. The first byte is the marker of the class; the following 3072 bytes are the pixel values of the image. The first 1024 bytes correspond to the red channel, the next 1024 bytes - to the green channel, the last 1024 bytes - to the blue channel. Values are arranged in rows, that is, the first 32 bytes are the values of the red channel of the first line of the image.

Thus, the experimental data consist of nearly 175MB of image data in binary format. The length of one image is 24576 bits. This value is the dimensionality of the SDM.

B. Computing Platform

NVIDIA CUDA, due to its cheap parallelism, is exceptionally suitable for implementing models like SDM [14].

$NVIDIA\ GeForce\ GTX\ 960M$ graphics card (Maxwell architecture [15]) was used to test the model. It employs 640 CUDA cores and $4GB$ of memory; its compatibility level is 5.0 CUDA compute. The calculations were performed on 64 one-dimensional thread blocks. Each block activated 512 threads (32768 threads were involved).

C. Feature-Based Encoding

Sparse representation of dense image data requires specific binary encoding. CIFAR-10 dataset is suitable for feature-based encoding by its design since it is a collection of images of different classes that barely intersect.

Sparse representations were obtained via Google Cloud Vision's API; this service provides cheap, fast, and efficient image classification and labeling. Only features with a level of significance higher than 0.7 were used, others were dropped.

The experiment employed $9,000$ encoded images; sparse representations with at least two features were selected for

further processing. The overall number of features (and the dimensionality of the model, respectively) is 651.

IV. Program Implementation

A. Basic Architecture

Essential reading/writing and storage operations were inherited from the implementation created for the previous research [4].

Sparse signal reconstruction problem required the following new modules:

- feature-matching activation function;
- CUDA kernel for basic threshold calculation;
- CUDA kernel for cellwise decision calculation in case of the "biological" model.

Also, initial memory allocation and mask generation procedures were adjusted to match new demands:

- increased number of cells;
- decreased dimensionality;
- a full set of pairs of indices for activation masks (instead of random mask generation).

The hybrid architecture framework developed during the research will be included as a component to the underlying software libraries for the SCIT cluster complex at V.M. Glushkov Institute of Cybernetics of NASU [16].

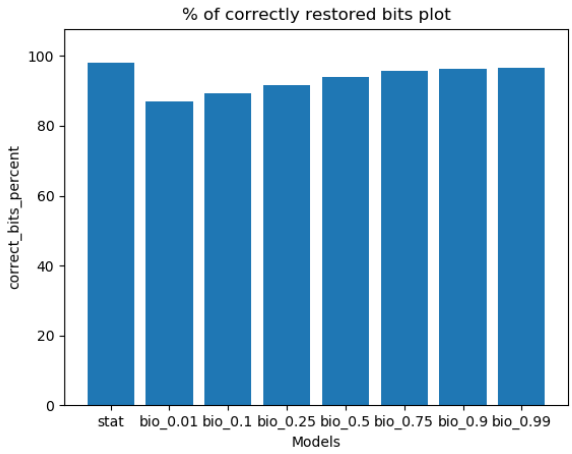

Fig. 1. Accuracy bar plot for a set of the tested models

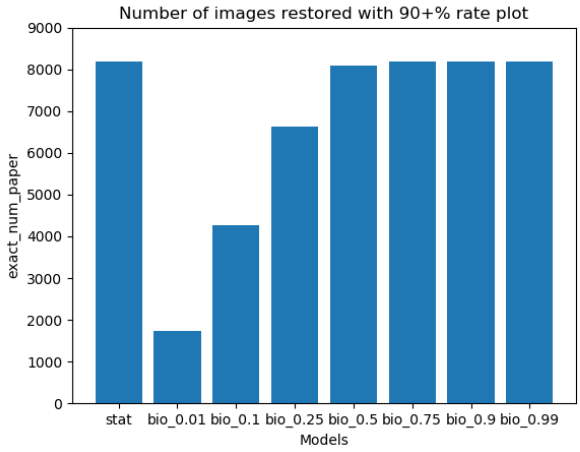

Fig. 2. Bar plot for the number of images for which features were reconstructed correctly with a 90% threshold.

V. Results of Simulations

A. Naïve Accuracy Estimation

The primary accuracy metric is the percent of correctly reconstructed bits (Fig. 1). The best model, in this case, is "statistical"; α parameter importance [17] for the "biological" model is also quite visible.

However, due to a vast 0/1 imbalance, naïve accuracy metric is not the right choice. Yet, it illustrates that low threshold values for the "biological" models do not give proper inference.

B. Threshold-based image restoration metric

Another option for accuracy estimation is to calculate the number of images for which the features were restored correctly with a certain threshold, 90% in our case (Fig. 2). This metric is more robust in case of a 0/1 imbalance, but it is not illustrative because most of the models perform similarly.

For this experiment, this metric shows that low threshold values for the "biological" models result in poor signal restoration quality for a general problem.

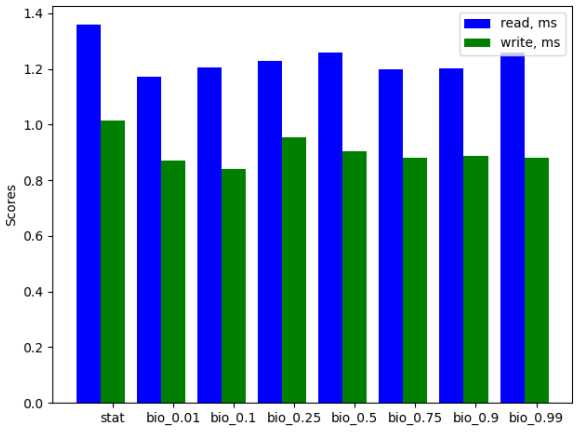

Fig. 3. Bar plot for reading/writing performance measurements for the examined models

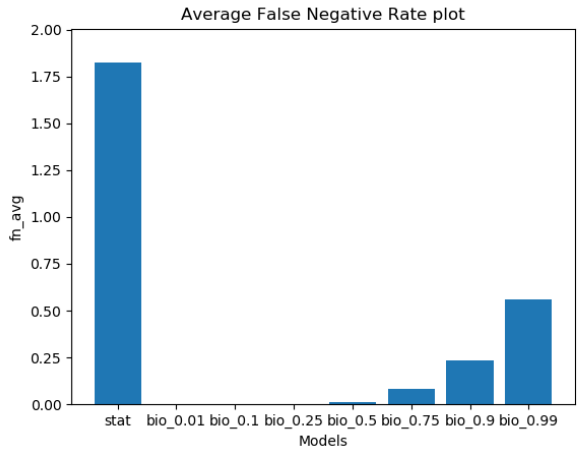

Fig. 4. Bar plot for the average false negatives.

C. Reading/Writing Operations Performance

As far as reading and writing operations are quite similar for all the observed models, and most of the calculations are shared, the performance measurements of the models are almost equal (Fig. 3).

It is worth mentioning that performance measurements for this experiment are close to the ones in a dense signal reconstruction experiment [4], though the dimensionality has reduced dramatically $24576 \rightarrow 651$. The reason for it is that at the same time, the number of memory cells has increased $(70000 \rightarrow 211575)$.

D. Average False Positives / False Negatives

Essential metrics for strongly imbalanced datasets are average false positives and average false negatives numbers since they can illustrate how good is the model in distinguishing "false" vs. "true."

Bar plot for the average false negatives (Fig. 4) shows that the "statistical" model is the worst one w.r.t recognizing 0's, but this fact is just one side of the coin since $f(x) \equiv 0$ would show the same result w.r.t. this metric.

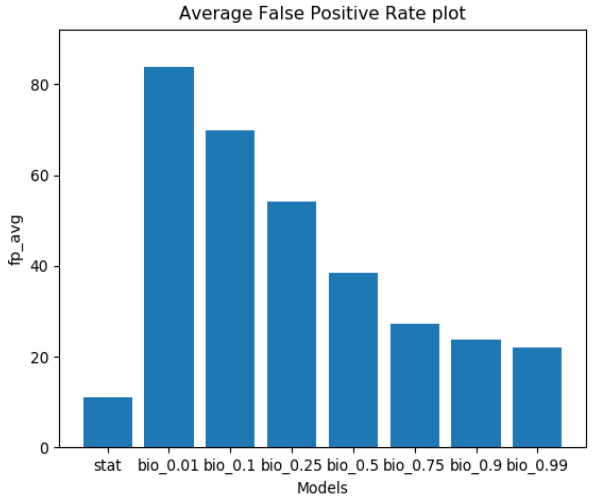

Fig. 5. Bar plot for the average false positives.

Bar plot for the average false positives (Fig. 5) shows that the "statistical" model vastly outperforms each of the parameterized "biological" family of models w.r.t. 1's recognition. It is also clear that increasing the "biological" threshold α leads to the growth of FN errors and the decrease of FP errors.

It is vital to notice that α parameter can be constructed adaptively. The respective algorithm must take into account partial $0/1$ probabilities for each physical memory cell, as well as some assumptions regarding the overall memory load level.

VI. Conclusion

A. Model Architecture

Several techniques for storing holistic structures in neural networks in a sparse representation have been proposed. It employs the Compressed Sensing approach along with a classic neural memory model.

The ensembled model is suitable for GPU parallelization, and reading/writing operations are fast even on old generation GPUs. Therefore, the model can be applied to solving numerous real-world problems, e.g., robotics, image recognition, etc.

B. "Statistical" vs. "Biological" Reading Methods

Two approaches for reading sparse data were suggested, implemented, and carefully examined. The "statistical" model averagely outperforms the parameterized family of "biological" ones. However, the overall error cost strongly depends on the general $0/1$ tradeoff for the particular problem.

C. Dense vs. Sparse Signal Reconstruction via SDM

The obtained results show that raw SDM reconstructs sparse signals less accurate than dense ones [18][19]. Proposed techniques for reading sparse data have significantly improved signal reconstruction error (w.r.t. average false positives/false negatives).

D. Future Plans

Further research plans are related to studying SDM applications for full-cycle signal compression and reconstruction.

References

[1] P. Kanerva, "Sparse Distributed Memory," MIT Press, Cambridge, MA. 1988.

[2] D. Smith, S. Forrest, and A. Perelson, "Artificial Immune Systems and Their Applications," pp. 105-112, Springer-Verlag, 1999.

[3] L. A. Jaeckel, "A Class of Designs for a Sparse Distributed Memory," Report RIACS TR 89.30, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1989.

[4] R. Vdovychenko, "Realizatsiya rozridzheno-rozpodilenoyi pamyati na suchasnykh hrafichnykh protsesorakh i doslidzhennya kharakterystyk modeli," [Implementation of Sparse Distributed Memory for modern GPU and model's features research], Komp'yuternaya Matematika, 2019.

[5] L. A. Jaeckel, "An Alternative Design for a Sparse Distributed Memory," Report RIACS TR 89.28, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1989.

[6] S. B. Furber, W. J. Bainbridge, J. M. Cumpstey, and S. Temple, "Sparse distributed memory using N-of-M codes," Neural Networks, Vol. 17, Issue 10, pp. 1437-1451, December 2004.

[7] G. Sjodin, "The Sparchunk Code: A Method to Build Higherlevel Structures in a Sparsely Encoded SDM," IEEE International Joint

Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence, 1998.

[8] H. D. Luke, "The Origins of the Sampling Theorem," IEEE Communications Magazine. 37 (4): 106–108, 1999.

[9] D. L. Donoho, "Compressed Sensing," IEEE transactions on information theory, 2006.

[10] E Candes, T. Tao, "Decoding by Linear Programming," EEE transactions on information theory, 2005.

[11] M. Lustig, D. L. Donoho, J. M. Santos, J. M. Pauly, "Compressed Sensing MRI," IEEE Signal Processing Magazine, 2008.

[12] A. Y. Yang et al., "Robust Face Recognition via Sparse Representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008.

[13] Y. Wiaux et al., "Compressed sensing imaging techniques for radio interferometry," Monthly Notices of the Royal Astronomical Society, 2009.

[14] M. S. Brogliato, D. M. Chada, and A. Linhares, "Sparse distributed memory: understanding the speed and robustness of expert memory," Frontiers in Human Neuroscience, April 2014.

[15] E. W. G. Clua, M. P. Zamith, "Programming in CUDA for Kepler and Maxwell Architecture," The Revista de Informtica Terica e Aplicada, the Institute of Informatics of the Federal University of Rio Grande do Sul, Vol. 22, No 2, 2015.

[16] A. Golovynskyi, I. Sergienko, V. Tul'chinskyi, et al. "Development of SCIT Supercomputers Family Created at the V. M. Glushkov Institute of Cybernetics, NAS of Ukraine, in 2002–2017". Cybernetics and Systems Analysis, 53(4), 600–604 (2017).

[17] J. J. Park, R. Boettcher, A. Zhao, A. Mun, K. Yuh, V. Kumar, and V. Marcolli, "Prevalence and Recoverability of Syntactic Parameters in Sparse Distributed Memories," Geometric Science of Information, pp. 265-272, 2017.

[18] M. J. Flynn, P. Kanerva, and N. Bhadkamkar, "Sparse Distributed Memory: Principles and Operation," Report CSL-TR-89-400, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1989.

[19] V. G. Tul'chinskii, I. N. Pshonkovskaya, and S. V. Zaytseva, "Fast retraining of SDM," Cybernetics and Systems Analysis. Vol. 35, No. 4, 1999.