Digital-based Processing In-Memory for Acceleration of Unsupervised Learning

Mohsen Imani[‡], Saransh Gupta[*], Yeseong Kim[*], Tajana Rosing[*]
[‡]Department of Computer Science, UC Irvine
[*]Department of Computer Science and Engineering, UC San Diego
Corresponding author: m.imani@uci.edu

I. Introduction

With the emergence of the Internet of Things (IoT), sensory and embedded devices generate massive data streams and demand services that pose huge technical challenges due to limited device resources. Today IoT applications analyze raw data by running machine learning algorithms. Since the majority of data generated are not associated with any labels, clustering algorithms are the most popular learning methods used for data analysis [1]. Clustering algorithms are unsupervised and have applications in many fields including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics [2]. These algorithms are used to group a set of objects into different classes, so that objects within the same class are similar to each other. The process of clustering datasets involves heavy computations as most algorithms need to calculate pairwise distances between all the points in the dataset.

Running clustering algorithms with large datasets on conventional processors results in high energy consumption and slow processing speed. Although new processor technology has evolved to serve computationally complex tasks more efficiently, data movement costs between the processor and memory still hinder the higher efficiency of application performance. Processing in-memory (PIM) is a promising solution to accelerate applications with a large amount of parallelism [3]. Several recent works have explored the advantage of PIM-based architectures to accelerate supervised learning algorithms such as Deep Neural Networks (DNNs) [4]. These approaches mostly use PIM architecture as a dot product engine to perform the vector-matrix multiplication involved in the DNN computation.

There are three main challenges in using existing PIM architectures to accelerate clustering algorithms: (i) the main operations involved in clustering algorithms are pairwise distance computation, e.g., Euclidean distance, and similarity search which cannot be supported entirely by existing PIM architectures [3]. (ii) Most existing PIM architectures are analog-based [4]; thus they use Digital-to-Analog Converter (DAC) blocks to transfer data to the analog domain for the computation and Analog-to-Digital Converter (ADC) to transfer it back to the digital domain. In the existing PIM architectures, the DAC/ADC blocks are dominating the total chip power/area, e.g., 98% of DNN accelerators [4], resulting in very low throughput/area. (iii) They require separate storage and computing memory units, resulting in a large amount of internal data movements. This not only reduces the computation efficiency but also affects the design scalability.

In this work, we present a digital-based PIM architecture, called DUAL, which accelerates a wide range of popular clustering algorithms on conventional crossbar memory. DUAL supports all essential clustering operations in memory, in a parallel and scalable way. DUAL eliminates the necessity of using any ADC/DAC blocks and addresses the internal data movement. The main contributions are listed as follows:

- To the best of our knowledge, DUAL **is the first digital-based processing in-memory architecture that accelerates unsupervised learning tasks.** In contrast to the existing PIM designs, DUAL enables all PIM computation on digital data stored in memory. This eliminates the necessity of using ADC/DAC blocks, providing high throughput/area. DUAL is also the first PIM architecture that support digital search-based Hamming distance computing.
- Instead of working on the original data, DUAL **maps all data points into high-dimensional space enabling the main clustering operations to process in a hardware-friendly way**. DUAL proposes a novel non-linear encoder that preserves the similarity of the neighbor values in high dimensional space. This encoding simplifies the distance similarity metric from Euclidean to Hamming distance.
- **We design a PIM architecture that accelerates various clustering algorithms on conventional crossbar memory.** DUAL performs in-place computation in a highly parallel and scalable way, where the data points can be processed without transferring between the storage and computing blocks. Therefore, it eliminates internal data movements between memory blocks. The proposed solution supports a wide range of essential clustering operations, e.g., in-memory distance computations and the nearest search, which can be programmed in high-level languages.

We have evaluated DUAL efficiency on several popular clustering algorithms and a wide range of large-scale datasets. Our evaluations show that DUAL provides a comparable quality of clustering to the baseline clustering algorithms. In terms of efficiency, DUAL provides 58.8× speedup and 251.2× energy efficiency improvement as compared to the state-of-the-art solution running on NVIDIA GTX 1080 GPU. Enabling 1% and 2% lower quality of clustering, DUAL speeds up the computation to 72.5× and 87.4× respectively.

II. DUAL Overview

In this paper, we propose DUAL, a novel platform to accelerate unsupervised learning in a fully digital PIM architecture. Figure 1a shows an overview of DUAL framework consisting of an HD-Mapper and a digital-based PIM accelerator. Instead of working on original data, our architecture maps all data points to long-size binary vectors. This data mapping replaces complex

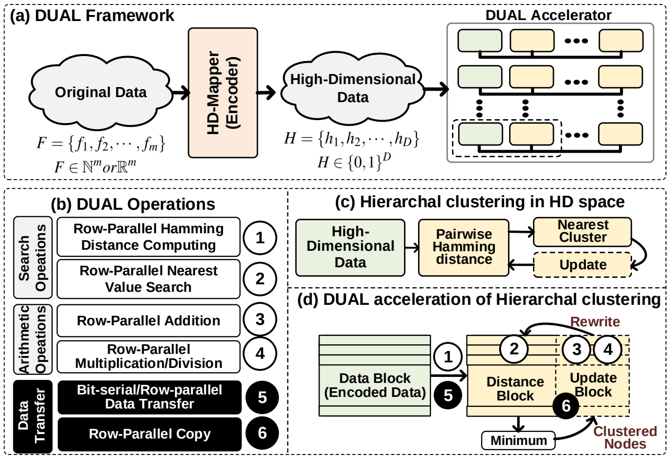
(a) DUAL Framework
Original Data
$F = \{f_1, f_2, \cdots, f_m\}$
$F \in \mathbb{N}^m or \mathbb{R}^m$
HD-Mapper (Encoder)
High-Dimensional Data
$H = \{h_1, h_2, \cdots, h_D\}$
$H \in \{0,1\}^D$
DUAL Accelerator
(b) DUAL Operations
Search Opeations
Row-Parallel Hamming Distance Computing
1
Row-Parallel Nearest Value Search
2
Arithmetic Opeations
Row-Parallel Addition
3
Row-Parallel Multiplication/Division
4
Data Transfer
Bit-serial/Row-parallel Data Transfer
5
Row-Parallel Copy
6
(c) Hierarchal clustering in HD space
High-Dimensional Data
Pairwise Hamming distance
Nearest Cluster
Update
(d) DUAL acceleration of Hierarchal clustering
Rewrite
Data Block (Encoded Data)
1
5
2
3
4
Distance Block
Update Block
6
Minimum
Clustered Nodes

Fig. 1. Overview of DUAL platform accelerating clustering algorithms.

clustering operations with hardware friendly operations. DUAL also exploits the resistive characteristics of Non-Volatile Memory (NVM), in particular memristor devices to support all necessary clustering operations in memory.

A. HD-Mapper

The goal of the HD-Mapper is to encode data points into high-dimensional vectors, called hypervector, such that the data can keep their similarity using a PIM-friendly Hamming distance metric. There are several approaches based on Hyperdimensional (HD) computing to perform the encoding functionality. However, all existing approaches linearly map each input feature into the hyperspace [5]–[7]. In contrast, we propose HD-mapper that explicitly considers non-linear interactions between input features. The proposed encoding is inspired by the Radial Basis Function (RBF) kernel trick method [8]. The underlying idea of HD-mapper is that data that is not linearly separable in original dimensions, might be linearly separable in higher dimensions.

Let us consider an encoding function that maps a feature vector $\mathbf{F} = \{f_1,\ f_2, \ldots,\ f_m\}$, with m features ($f_i \in \mathbb{R}$) to a hypervector $\mathbf{H} = \{h_1,\ h_2, \ldots,\ h_D\}$ with D dimensions ($h_i \in \{0,1\}$). We generate each dimension of encoded data by calculating a dot product of feature vector with a randomly generated vector as $h_i = cos(\mathbf{B}_i \cdot \mathbf{F})$, where B_i is a randomly generated vector from a Gaussian distribution (mean $\mu = 0$ and standard deviation $\sigma = 1$) with the same dimensionality of the feature vector. The random vectors $\{\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_D\}$ can be generated once offline and then can be used for the rest of the classification task ($\mathbf{B}_i \in \mathbb{R}^m$). After this step, each element, h_i of a hypervector $\mathbf{H}$, has a non-binary value. We prefer binary hypervectors for computation efficiency.

B. DUAL Accelerator

The second module is a digital-based PIM architecture that enables parallel encoding and clustering computation over the encoded hypervectors stored in memory. Unlike prior PIM designs that use large ADC/DAC blocks for analog computing [4], DUAL performs all clustering computations on the digital data stored in memory. This eliminates ADC/DAC blocks, resulting in high throughput/area and scalability. DUAL uses two blocks for performing the computation; a *data block* and a *distance block*. The data block stores the encoded data points and computes pairwise similarity using a row-parallel Hamming distance computation. Each distance/data block supports the following set of operations (shown in Figure 1b): (i) *search-based operations*: row parallel Hamming distance computation and nearest search. (ii) *Arithmetic operations*: row-parallel addition, multiplication and division.

Figure 1c,d shows how DUAL maps hierarchical clustering into PIM acceleration. In each iteration, DUAL computes the Hamming distance of each data point with all stored data points in all data blocks using the row-parallel search operation and the result is written in a *distance memory*. After computing all pairwise distances, DUAL performs the search for the nearest value over the distance matrix. Our design supports the nearest search operation in a row-parallel way. Next, DUAL clusters the two data points with the highest similarity and then updates the relative distance of all other data points with the clustered nodes. The distance update is computed using linear arithmetic operations, e.g., addition, multiplication, which can be performed in a row-parallel way in the *update block* (Figure 1c,d). The updated distance vectors will be written back into the corresponding row/column of the *distance block*. DUAL continues computation by iteratively finding and clustering data points with the closest distance. DUAL exploits the supported PIM operations to perform clustering tasks where data is already stored in memory. DUAL also uses interconnects to enable bit-serial/row-parallel data transfer between the data and distance blocks. This eliminates the overhead of internal data movement between the data and distance blocks (Figure 1c,d).

Acknowledgements

This abstract is based on our accepted paper in 53rd IEEE/ACM International Symposium on Microarchitecture (MICRO), 2020 [9].

References

[1] D. Singh and C. K. Reddy, “A survey on platforms for big data analytics,” *Journal of big data*, vol. 2, no. 1, p. 8, 2015.

[2] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[3] M. Imani, S. Gupta, Y. Kim, and T. Rosing, “Floatpim: In-memory acceleration of deep neural network training with high precision,” in *Proceedings of the 46th International Symposium on Computer Architecture*, pp. 802–815, ACM, 2019.

[4] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, “Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars,” *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.

[5] A. Cano, N. Matsumoto, E. Ping, and M. Imani, “Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2021.

[6] M. Imani *et al.*, “Exploring hyperdimensional associative memory,” in *HPCA*, pp. 445–456, IEEE, 2017.

[7] P. Poduval, Z. Zou, X. Yin, E. Sadredini, and M. Imani, “Cognitive correlative encoding for genome sequence matching in hyperdimensional system,” in *IEEE/ACM Design Automation Conference (DAC)*, 2021.

[8] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *Advances in neural information processing systems*, pp. 1177–1184, 2008.

[9] M. Imani, S. Pampana, S. Gupta, M. Zhou, Y. Kim, and T. Rosing, “Dual: Acceleration of clustering algorithms using digital-based processing in-memory,” in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 356–371, IEEE, 2020.