Flexible and Personalized Learning for Wearable Health Applications using HyperDimensional Computing

Sina Shahhosseini, Yang Ni, Emad Kasaeyan Naeini, Mohsen Imani, Amir M. Rahmani, Nikil Dutt
University of California Irvine, Irvine,USA
{sshahhos,yni,ekasaeya,m.imani,a.rahmani,dutt}@uci.edu

ABSTRACT

Health and wellness applications increasingly rely on machine learning techniques to learn end-user physiological and behavioral patterns in everyday settings, posing two key challenges: inability to perform on-device online learning for resource-constrained wearables, and learning algorithms that support privacy-preserving personalization. We exploit a Hyperdimensional computing (HDC) solution for wearable devices that offers flexibility, high efficiency, and performance while enabling on-device personalization and privacy protection. We evaluate the efficacy of our approach using three case studies and show that our system improves performance of training by up to 35.8× compared with the state-of-the-art while offering a comparable accuracy.

CCS CONCEPTS

• **Computer systems organization** → *Embedded software.*

KEYWORDS

HyperDimensional Computing, Wearable Devices, Personalization

ACM Reference Format:

Sina Shahhosseini, Yang Ni, Emad Kasaeyan Naeini, Mohsen Imani, Amir M. Rahmani, Nikil Dutt. 2022. Flexible and Personalized Learning for Wearable Health Applications using HyperDimensional Computing. In *Proceedings of the Great Lakes Symposium on VLSI 2022 (GLSVLSI '22), June 6–8, 2022, Irvine, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3526241.3530373

1 INTRODUCTION

Wearable devices play a significant role in health monitoring systems by continuously monitoring human physiological and physical data [1]. Health and wellness applications increasingly rely on machine learning (ML) algorithms to capture the user's behavioral and physiological patterns [2], but pose two challenges: 1) inability to perform on-device learning for resource-constrained wearables, as ML workloads require significant computational power and storage [3, 4], and 2) the need to develop on-device and online learning algorithms that support privacy-preserving personalization. Nowadays, the majority of wearable devices (e.g., smart watches) are multi-application and capable of interacting with users to collect feedback and personalize their models over time to meet the unique characteristics of each person, however, online training of ML models on wearable devices currently is not feasible using state-of-the-art (SOTA) machine learning algorithms [5]. For these reasons, alternative learning algorithms are required to deliver real-time, low-power, and personalize services on wearable devices, and therefore, it is desired to re-design the learning process with respect to both algorithm and hardware.

Hyperdimensional computing (HDC) offers an alternative computational [6, 7]. HDC is based on the understanding that the human brain operates on *high-dimensional* representations. HDC offers a well-suited solution for online learning and personalization on resource-constrained devices since: (i) HDC models are computationally efficient, highly parallel to train, and amenable to hardware-level optimization [8, 9]. (ii) HDC can naturally enable on-device online learning for wearable devices [10], thereby facilitating privacy-preservation and personalization [11]. There have been recent efforts [12–16] to deploy HDC algorithms to offer efficient on-device learning using single-pass training. However, single-pass training provides very weak classification accuracy compared to online learning in HDC [10].

Furthermore, recent wearable solutions mainly focus on monitoring vital signs, which has limitations for wellness prediction. The data obtained from various signal sources might span multiple dimensions across multiple scales and exhibit varying precision. Some features may emerge due to users' behavioral patterns and context, which will impose a much higher degree of variation from one user to another user. These variations result in degrading the performance of general wearable health applications. Thus, it is imperative to integrate personalized health-related data collected from various sources. Existing on-device learning solutions [13–17] fail to offer personalized learning for wearable devices. In addition, these solutions are targeted for specific applications and platforms limiting their utility for multi-purpose wearable devices. For example, Moin et al. [16] proposed a custom gesture recognition system, while Bhat et al. [17] proposed a DNN-based activity recognition system based on Application Specific Integrated Circuit (ASIC) platform, making the solutions application-specific. Therefore, The literature lacks a comprehensive solution which is *flexible* and *platform-agnostic* to run a variety of health applications, in particular on multi-purpose wearables.

In this paper, we propose a flexible and personalized HDC-based learning approach for wearable devices running health applications. Our approach enables accurate online on-device training and avoids model saturation by adopting and customizing the HDC training strategy presented in [10] for resource-constrained wearable devices. In summary, our main contributions are:

- We implement an online training framework for efficient and accurate learning using HDC algorithms on both CPU and FPGA

GLSVLSI '22, June 6–8, 2022, Irvine, CA, USA.

ACM ISBN 978-1-4503-9322-5/22/06.
https://doi.org/10.1145/3526241.3530373

platforms. We present the HDC framework as a practical and flexible solution for efficient on-device and online learning for wearable applications. We also demonstrate that on-device learning enables *personalization* and *user privacy protection* for wearable devices. The CPU implementation of our approach can be readily implemented on existing off-the-shelf multi-purpose wearable devices such as smartwatches.

- We demonstrate the effectiveness of our solution using three case studies compared with SOTA learning algorithms. Our evaluation shows that our HDC-based system improves training performance of the wearables by up to 35.8× compared with DNN while providing comparable accuracy.

2 HYPERDIMENSIONAL CLASSIFICATION

We present a robust and lightweight hyperdimensional classification. The first step in HDC is to encode data into a high-dimensional space. Then, HDC performs a learning task over encoder data by performing a single-pass training that generates a hypervector representing each class. The inference task can performed by checking the similarity of an encoded query to the class hypervector. Let's assume $\vec{\mathcal{H}}_1$, $\vec{\mathcal{H}}_2$ are two randomly generated hypervectors ($\vec{\mathcal{H}} \in \{-1,+1\}^D$) and $\delta(\vec{\mathcal{H}}_1, \vec{\mathcal{H}}_2) \approx 0$. HDC is based on a set of primitives: **(1) Bundling:** is an addition of multiple hypervectors into a single hypervector, $\vec{\mathcal{R}} = \vec{\mathcal{V}}_1 + \vec{\mathcal{V}}_2$, where $\vec{\mathcal{V}}_2 \in \{0,1\}^D$ and D is the dimension of the HDC space. Unlike original space where bundling act as an average operation, in high-dimensional space the addition is memorization function. **(2) Binding:** associates multiple orthogonal hypervectors (e.g., $\vec{\mathcal{V}}_1$, $\vec{\mathcal{V}}_2$) into a single hypervector ($\vec{\mathcal{R}} = \vec{\mathcal{V}}_1 * \vec{\mathcal{V}}_2$). The bound hypervector is a new object in HDC space which is orthogonal to all input hypervectors ($\delta(\vec{\mathcal{R}}, \vec{\mathcal{V}}_1) \simeq 0$ and $\delta(\vec{\mathcal{R}}, \vec{\mathcal{V}}_2) \simeq 0$). **(3) Permutation:** defined as a single rotational shift. The permuted hypervector will be nearly orthogonal to its original hypervector ($\delta(\vec{\mathcal{V}}_1 \rho \vec{\mathcal{V}}_1) \simeq 0$).

2.1 Online and Iterative Learning

We propose an adaptive training framework for efficient and accurate learning in HDC. Our training identifies common patterns during training and eliminates the saturation of the class hypervectors during traditional single-pass training. Instead of naively combining all encoded data points, our approach adds each encoded data to class hypervectors depending on how much new information the pattern adds to class hypervectors. If a data point already exists in a class hypervector, the framework will add no or a tiny portion of data to the model to eliminate hypervector saturation. Figure 1a shows the framework's functionality during adaptive initial training. Let's assume $\vec{\mathcal{H}}$ as a new training data point. The framework computes the cosine similarity of $\vec{\mathcal{H}}$ with a class hypervector that has the same label as $\vec{\mathcal{H}}$. If the data point corresponds to i^{th} class, we compute similarity of a data point with $\vec{C}_i$ as: $\delta(\vec{\mathcal{H}}, \vec{C}^l) = \frac{\vec{\mathcal{H}} \cdot \vec{C}^l}{\|\vec{\mathcal{H}}\| \cdot \|\vec{C}^l\|}$ where $\vec{\mathcal{H}} \cdot \vec{C}^l$ is a dot product between a query and class hypervector (A). The δ value shows the similarity of a data point to its class hypervector. Instead of naively adding data point to the model, the framework updates the model based on the δ similarity. For example, if an input data has label l,

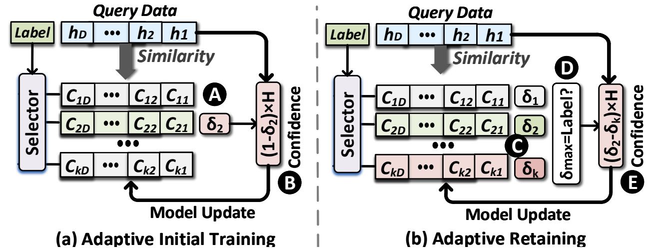


Figure 1: (a) HDC Online Learning (b) Iterative Learning.

the model updates as follows (B).

$$\vec{C}_l \leftarrow \vec{C}_l + \eta\ (1 - \delta_l) \times \vec{\mathcal{H}} \tag{1}$$

where η is a learning rate. A large δ_l indicates that the input is a common data point which is already exist in the model. Therefore, our update adds a very small portion of encoded query to model to eliminate model saturation ($1 - \delta_l \simeq 0$). However, small δ_l means that the query has new pattern which does not exist in the model. Thus, the model is updated with a larger factor ($1 - \delta_l \simeq 1$).

Although single-pass training is suitable for fast and ultra-efficient learning, embedded devices may have enough resources to perform more accurate learning tasks. Our framework supports retraining to enhance the quality of the model. Instead of starting to retrain from a naive initial model, the framework retraining starts from the initial adaptive model (explained in Section 2.1). The framework's initial model already considered the weight of each input data during single-pass training. Therefore, the framework retraining starts from a well-trained initial model with relatively high classification accuracy. This enables the framework to retrain the model with a much lower number of iterations, resulting in fast convergence. Figure 1b shows the framework functionality during adaptive retraining. the framework follows a similar learning procedure as initial training. For each training data point, say H, the framework checks the similarity of data with all class hypervectors in the model (C) and updates the model for each miss-prediction (D). Retraining examines if the model correctly returns the label l for an encoded query $\vec{\mathcal{H}}$. If the model mispredicts it as label l', the model updates as follows (E).

$$\begin{aligned} \vec{C}_l &\leftarrow \vec{C}_l + \eta\ (\delta_{l'} - \delta_l) \times \vec{\mathcal{H}} \\ \vec{C}_{l'} &\leftarrow \vec{C}_{l'} - \eta\ (\delta_{l'} - \delta_l) \times \vec{\mathcal{H}} \end{aligned} \tag{2}$$

where $\delta_l = \delta(H, \vec{C}_l)$ and $\delta_{l'} = \delta(H, \vec{C}_{l'})$ are the similarity of data with correct and miss-predicted classes, respectively. This ensures that we update the model based on how far a train data point is miss-classified with the current model. In case of of a very far miss-prediction, $\delta_{l'} >> \delta_l$, the framework retraining makes a major changes on the mode. While in case of marginal miss-prediction, $\delta_{l'} \simeq \delta_l$, the update makes smaller changes on the model.

2.2 HDC for Wearable Devices

Figure 2 shows our proposed monitoring system. The system collects the raw data from sensors such as photoplethysmography (PPG), etc. Self-reported labels are collected through the wearable's

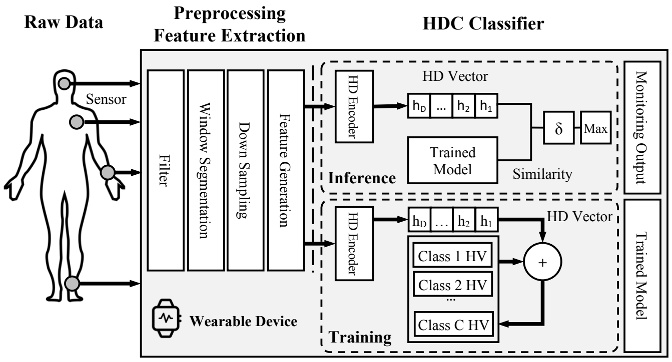


Figure 2: HDC for Health monitoring system architecture.

user interface (e.g., an smartwatch's touchscreen) from subjects in-the-moment to personalize the model for each individual over time. The collected sensory data is processed through two major steps at the wearable device: **(a) Preprocessing and Feature Extraction**, **(b) HDC Classifier**. The preparation includes data integration, data cleaning, and data reduction. Feature generation is the next step which derives values intended to be informative and non-redundant. It facilitates more accurate subsequent learning. The HDC classifier is the final step where the training and inference phase are performed. During the training, our framework provides single-shot and iterative training known as HD-Online and HD-Iterative, respectively. HD-Online identifies common patterns within a single-pass training. This results in learning the model as input data comes from the sensors. On the other hand, HD-Iterative offers retraining to enhance the quality of the model.

3 EVALUATION AND ANALYSIS

We implement the HD-based monitoring system on two embedded platforms: Raspberry Pi 3B+ using ARM CPU and Xilinx Kintex 7 FPGA. We implement HD functionality on FPGA using Verilog and synthesize it based on a SOTA FPGA framework [7]. We demonstrate the effectiveness of our proposed system with three case studies, including HAR, PM, and SM applications. In the following, we evaluate our proposed system's accuracy, performance, and energy efficiency on these CPU and FPGA. We compare HD algorithm accuracy against SOTA learning algorithms, including Deep Neural Network (DNN), Support Vector Machine (SVM). DNN and SVM models are trained with Tensorflow and the Scikit-learn library, respectively. We use the common practice of the grid search to identify the best hyperparameters for each model. The neural network architecture consists two hidden layers with 512, and 128 neurons. The HD algorithm is trained using a single-pass (HD-Online) and iterative (HD-Iterative) way using $D = 4k$. We evaluate our proposed system using three health case studies: Pain Monitoring (PM), Stress Monitoring (SM), Human Activity Recognition (HAR).

3.1 Accuracy Analysis

In this subsection, we demonstrate the prediction accuracy of the HD algorithm through experimental comparisons against SOTA learning algorithms. We demonstrate how personalization can be

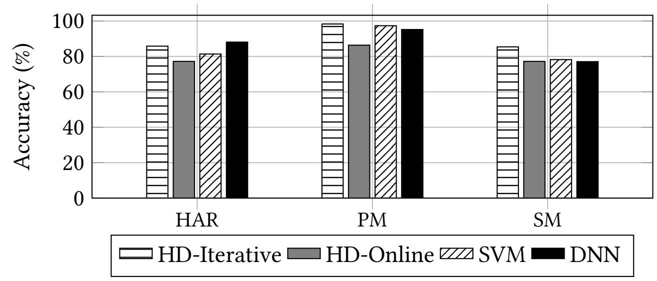


Figure 3: HD vs SOTA accuracy analysis.

Table 1: Accuracy Analysis for the Personalized model vs. the General model for the Stress monitoring application.

Model	Strategy	S1	S2	S3	S4	S5	S6
Pers.	**Online**	77.2 %	78.2%	79.2%	64.6%	68.5%	64.9%
	Iterative	85.4%	87.4%	88.2%	79.6%	82.9%	79.6%
Gen.	**Online**	50.6%	49.9%	51.8%	48.3%	50.1%	59.0%
	Iterative	72.6%	62.9%	73.6%	74.5%	74.6%	76.6%

achieved using the HDC-based classifiers. We first train the models based on the collected data from all subjects. Figure 3 shows that the HD approach provides comparable accuracy to the SOTA learning algorithms for three health monitoring applications. We report the results for HD algorithm for both iterative and online strategies. The HD-Iterative results show errors of 0.03% and 0.02% in comparison with DNN the algorithm for the HAR and Stress Monitoring applications, respectively. However, HD-iterative is even 0.1% more accurate than DNN algorithm for the PM application. On the other hand, the HD-Online method leads to 6.9%, 8.8%, and 2.7% accuracy degradation compared with the DNN algorithm for HAR, PM, and SM, respectively. Our evaluation shows HD-iterative leads to 4.4%, 0.1% accuracy improvement compared with SVM learning algorithm for HAR and PM application, respectively. While, HD-Online results in 2%, 8% accuracy error in comparison with SVM algorithm.

The bias in physiological data can be different for personal or general dataset [1]. We report the effect of personalization and how it improves the monitoring accuracy. We evaluate the personalization considering six participants (S1-S6) for the Stress monitoring application. To train the *General* model, we exclude the data from one subject and then train the model using data from all other subjects. We test the model on half of the data from the excluded subject (selected randomly). To train the *Personalized* model, we use the first half of each subject's data for training (to emulate the progression of time) and then test it with the second half of the subject's data. Table 1 shows *Personalized* model performance in comparison with *General* model. The results show an average of 20.48% and 11.38% improvement on accuracy when personalization is used for both HD-Online and HD-Iterative, respectively.

3.2 Performance Analysis

We report the performance of the HD learning algorithm during the training and inference phase on the platforms mentioned above. We

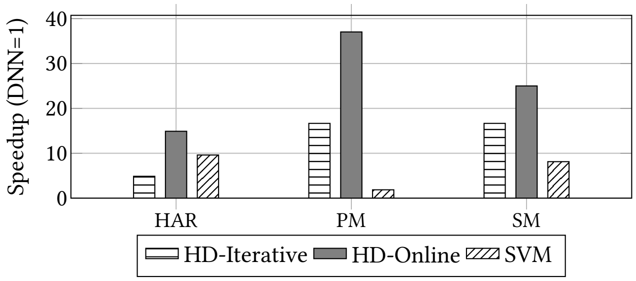
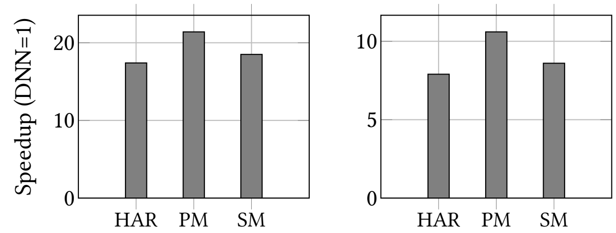


Figure 4: Training time for HD vs. SOTA on CPU.



(a) Training **(b) Inference**

Figure 5: Performance of HD-iterative vs. DNN on FPGA.

evaluate three health monitoring applications, including HAR, PM, and SM, with HD and SOTA learning algorithms. Figure 4 shows that HD algorithm provides significant speedup for training time compared with other algorithms. HD-Iterative algorithm significantly reduces training time by 4.8×, 15.3×, and 15.8× compared with the DNN algorithm for HAR, PM and SM applications, respectively. On the other hand, HD-Online algorithm results in better training time where it provides 14.6×, 35.8×, and 23.81× compared with DNN algorithm for HAR, PM, and SM applications, respectively. This speedup comes from HD-Online algorithm capability in lowering number of required training iterations. Table 2 also compares HD inference and training phase performance with the SOTA learning algorithms. HD algorithm presents 32×, 43.5×, and 4.29× improvement in inference time compared with SVM learning algorithm for HAR, PM, and SM, respectively. In addition, Figure 5 shows performance evaluation for HD algorithm during training and inference on the FPGA platform. HD algorithm can achieve up to 21.4× and 10.6× speedup during training and inference, respectively. Comparing the results between the FPGA and CPU platforms shows the FPGA design significantly improves the performance. mainly because CPUs use the same number of resources to perform 1-bit or 8-bit arithmetic operations, which limits the degree of parallelism in the CPU [7]. In contrast, FPGAs are significantly efficient for implementing low-precision arithmetic operations [7].

4 CONCLUSIONS

We proposed an adaptive HDC training framework for health monitoring systems that achieves fast, energy-efficient, and accurate on-device training/inference, and also enables personalization and privacy protection for wearable devices. We demonstrated the efficacy of our HDC approach using three realistic wearable healthcare studies, achieving better performance for training by up to 35.8× compared to state-of-the-art DNN algorithms while achieving comparable accuracy. We believe that our HDC-based framework is a promising approach to meet the low-power, personalization, and privacy requirements for health monitoring applications.

Table 2: Performance analysis HD vs. SOTA on CPU.

		Execution Time (sec)			
		HD-Iterative	**HD-Online**	**SVM**	**DNN**
HAR	Training	3.10	1.02	1.58	15.10
	Inference	0.01	0.01	0.32	0.01
PM	Training	3.59	1.54	29.67	55.25
	Inference	0.08	0.08	3.48	0.35
SM	Training	1.89	1.26	3.74	30.17
	Inference	0.97	0.98	4.21	0.45

ACKNOWLEDGEMENTS

This work was supported in part by National Science Foundation (NSF) #2127780, Semiconductor Research Corporation (SRC) Task #2988.001, Department of the Navy, Office of Naval Research, grant #N00014-21-1-2225 and #N00014-22-1-2067, Air Force Office of Scientific Research, and a gift from Cisco.

REFERENCES

[1] Edward Sazonov. *Wearable Sensors: Fundamentals, implementation and applications.* 2020.
[2] Trishan Panch et al. Artificial intelligence, machine learning and health systems. *Journal of global health*, 2018.
[3] Marcelo Brandalero et al. Multi-target adaptive reconfigurable acceleration for low-power iot processing. *IEEE TC*, 2020.
[4] Dolly Sapra et al. Constrained evolutionary piecemeal training to design convolutional neural networks. In *Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2020.
[5] Sauptik Dhar et al. On-device machine learning: An algorithms and learning theory perspective. *arXiv preprint arXiv:1911.00623*, 2019.
[6] Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. 2009.
[7] Mohsen Imani et al. Revisiting hyperdimensional learning for fpga and low-power architectures. In *HPCA*. IEEE, 2021.
[8] Behnam Khaleghi et al. tiny-hd: Ultra-efficient hyperdimensional computing engine for iot applications. In *DATE*, 2021.
[9] Mohsen Imani et al. Exploring hyperdimensional associative memory. In *HPCA*, 2017.
[10] Hernandez-Cane et al. Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system. In *DATE*, 2021.
[11] Sahand Salamat et al. Accelerating hyperdimensional computing on fpgas by exploiting computational reuse. *IEEE Transactions on Computers*, 2020.
[12] Abbas Rahimi et al. Hyperdimensional computing for noninvasive brain-computer interfaces: Blind and one-shot classification of eeg error-related potentials. In *BICT*, number CONF, 2017.
[13] Abbas Rahimi et al. Hyperdimensional computing for blind and one-shot classification of eeg error-related potentials. *Mobile Networks and Applications.*
[14] Simone Benatti et al. Online learning and classification of emg-based gestures on a parallel ultra-low power platform using hyperdimensional computing. *IEEE transactions on biomedical circuits and systems*, 13(3):516–528, 2019.
[15] Alessio Burrello et al. An ensemble of hyperdimensional classifiers: Hardware-friendly short-latency seizure detection with automatic ieeg electrode selection. *IEEE journal of biomedical and health informatics*, 25(4):935–946, 2020.
[16] Ali Moin et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nature Electronics*, 4(1):54–63, 2021.
[17] Ganapati Bhat et al. An ultra-low energy human activity recognition accelerator for wearable health applications. *ACM TECS*, 2019.