EnHDC: Ensemble Learning for Brain-Inspired Hyperdimensional Computing

Ruixuan Wang, Dongning Ma, Graduate Student Member, IEEE, and Xun Jiao, Member, IEEE

Abstract—Recently, brain-inspired hyperdimensional computing (HDC) becomes an emerging computational scheme that has achieved success in various domains, such as human activity recognition, voice recognition, and bio-medical signal classification. HDC mimics the brain cognition and leverages high-dimensional vectors (e.g., 10,000 dimensions) with fully distributed holographic representation and (pseudo-)randomness. Ensemble learning is a classical learning method utilizing a group of weak learners to form a strong learner, which aims to increase the accuracy of the model. This letter presents a systematic effort in exploring ensemble learning in the context of HDC and proposes an ensemble HDC model referred to as EnHDC. EnHDC uses a majority voting-based mechanism to synergistically integrate the prediction outcomes of multiple base HDC classifiers. To enhance the diversity of base classifiers, we vary the encoding mechanisms, dimensions, and data width settings among base classifiers. By applying EnHDC on a wide range of applications, results show that EnHDC can achieve on average 3.2% accuracy improvement over a single HDC classifier. Further, we show that EnHDC with reduced dimensionality can achieve similar or even surpass the accuracy of baseline HDC with higher dimensionality. This leads to a 20% reduction of storage requirement of the HDC model, which can enhance the efficiency of HDC enabled on low-power computing platforms.

Index Terms—Ensemble learning, hyperdimensional computing (HDC), memory-efficient computing.

I. INTRODUCTION

INSPIRED by how the human brain functions, hyperdimensional computing (HDC) is an emerging computing scheme that leverages the abstract patterns and mathematical properties of vectors in high dimension spaces [13], [14]. Rather than processing actual numbers, HDC works with hypervectors (HVs), which are high dimensional (e.g., 10,000 dimensions), holographic (not micro-coded) vectors with independent and identically distributed (i.i.d.) elements [9]. As a novel computing scheme, HDC has shown promising performance for various applications, such as natural language processing [13], voice recognition [7], and bioinformatics [12]. Compared with traditional computing schemes, such as neural networks, HDC has several advantages, such as a smaller model and lower computing cost, making it a promising computing scheme with low-power computing platforms and edge computing devices [8]. In addition, the memory-centricity of HDC grants the advantage of embracing the emerging energy-efficient in-memory computing schemes over other machine learning algorithms such as neural networks [10].

Ensemble learning is a machine learning paradigm where multiple models (often called “weak learners”) are trained to solve the same problem and combined to get better results [4]. Typically, an ensemble learning system aims to improves the performance by combining diverse weak learners (base classifiers). Using an ensemble model that combines the output from several models, e.g., averaging them, can reduce the risk of an unfortunate selection of a particularly poor classifier. For HDC, prior works inspire that a collection of HDC classifiers can achieve comparable performance for specific (e.g., EEG) applications with low overhead [3]. However, such a method still relies on external architectures such as linear layers that requires additional training effort for the aggregation from the results output by a single HDC model. Building an ensemble classifier using HDC itself under different configurations and (hyper-)parameters as well as the design exploration still require further study.

This letter explores the use of ensemble learning on HDC models and develop the ensemble HDC classifier, EnHDC for various applications. Leveraging the diversity of base classifiers, the ensemble classifier formulated by EnHDC achieves improved accuracy and reduced model size without additional architectures such as post-processing linear layers that requires training. We make the following contributions.

1) By leveraging the aggregated intelligence of a variety of HDC classifiers with different random initialization, EnHDC is able to achieve on average 3.2% accuracy improvement over a single HDC classifier.
2) We diversify of base classifiers by varying a diverse set of parameters, such as number of dimensions, data width, and encoding methods. This further leads to 1.2% accuracy improvement over basic EnHDC classifier.
3) We evaluate the EnHDC on four different practical application domains, including image classification, human activity recognition, speech recognition, and medical diagnosis. EnHDC enables smaller HDC models of about 20% size reduction with no accuracy drop.

II. HDC PRELIMINARIES

A. Notions in HDC

HV is a type of high-dimensional, holographic vectors with i.i.d. elements [9]. An HV of n dimensions can be noted by \( \hat{H} = (h_1, h_2, \ldots, h_n) \), where \( h_i \) denotes the elements inside the HV. HVs use their high dimensional space to store different layers of information, thus can represent values, features,
and even data samples. To establish the dynamic connection between different layers of information in HV, methodologies of aggregating information from HVs such as HDC operations, are therefore necessary.

HVs support three basic operations: 1) addition (+); 2) multiplication (*); and 3) permutation (ρ). Additions and multiplications take two HVs as input and perform element-wise operations that add or multiply each element inside the HVs index by index. Permutations only take one HV and perform cyclic shift over the HV. For all the three operations, the input HVs and the output HVs are in the same dimension. Addition is used to aggregate parallel features that usually belongs to one modality, while multiplication is used to combine different types of features together to create new features. Permutation is used to reflect spatial or temporal changes in the features.

Similarity check is used in HDC for the objective of measuring the similarity δ of information between different HVs. There are different algorithms to measure similarity, such as the Euclidean distance and Hamming distance, while in EnHDC, we are using cosine similarity as noted by

\[ \delta(H_q, H) = \frac{\langle H_q, H \rangle}{||H_q|| \times ||H||} \]

A higher similarity between two HVs indicates that they share more alike information, or vice versa.

B. HDC in Learning Tasks

HDC in learning tasks features three major phases: 1) Encoding; Training; and 3) Inference.

Encoding is the process of mapping input features of one sample to the high-dimensional space available for HDC training, inference, i.e., building representative HVs of a sample from the fundamental item memory using combinations of HDC operations. Item memory is a type of specially allocated memory during runtime, which stores the bottom layer HVs that are used to establish other HVs. To ensure the i.i.d. property, HVs in the item memories are all randomly initialized. Assume we have the m-dimensional input features \( \vec{F} = (f_1, f_2, \ldots, f_m) \) for each sample, a set of corresponding item memories \( \mathcal{R} = \{R_1, R_2, \ldots, R_m\} \) and the combination of HDC operations \( E \) determined by the application, the encoded HV \( \vec{H} \) is obtained by looking up each feature’s corresponding HV in the item memory and then applying them into the HDC operation combination:

\[ \vec{H} = E(\mathcal{R}, \vec{F}) = E(R_1[f_1], R_2[f_2], \ldots, R_m[f_m]). \]

The encoded HV will subsequently represent the input sample in training and inference.

Training is the process of aggregating encoded HVs sharing the same label to build the associative memory. Associative memory stores the class HVs, each representing a class in the learning problem. Training can be denoted as \( \mathcal{A} = \{\sum H_1^k, \sum H_2^k, \ldots, \sum H_k^k\} \). Assume we have a learning problem with \( k \) classes, and the encoded HVs \( \vec{H}_l \) for each training sample where \( l \) means the class label, training process to establish the associative memory \( \mathcal{A} \) is by summing HVs representing samples from the same class in the training set.

Inference is the process of using the associative memory established in the training phase to determine the class label of an unknown sample. Inference can be denoted as

\[ l = \arg \max (\delta(H_q, A^1), \delta(H_q, A^2), \ldots, \delta(H_q, A^k)) \]

First, we encode the unknown sample into its representing HV \( \vec{H}_q \) referred to as the query HV. Then, we perform similarity check between the query HV and each class HV inside the associative memory. As aforementioned, higher similarity means higher common information shared by the two HVs, further indicating that these two HVs are likely from the same class. Therefore, the class of HVs in the associative memory having the highest similarity is determined to be the class of the query HV, namely, the predicted label of the sample.

III. EnHDC Model

In this section, we describe the development of EnHDC and the enhancement to the diversity of the base classifiers in EnHDC for further performance improvement. Fig. 1 is the overview of EnHDC. First, we separately train several different base classifiers using different parameter configurations. We subsequently integrate these base classifiers into one ensemble classifier. Then, we encode the testing sample into query HV and perform inference on every base classifier. As we collect all the base inference results, we employ majority voting to obtain the inference result of the ensemble classifier.

A. Base Classifier Development

In traditional ensemble learning, base classifiers are developed with different initialization settings. In HDC, base classifiers are developed as described in Section II-B. Each base classifier has different configurations and different randomly generated item memories \( \mathcal{R} \). Thus, the training outcome, i.e., the associate memory, will have different class HVs representing the same class. Therefore, for a given query input, the classification output may be different.

B. Diversity Enhancement of Base Classifier

As the performance of base classifiers can be subpar, we propose to formulate the ensemble classifier by diversifying the base classifiers with different configurations, including encoding mechanisms, dimensions, and data widths. Specifically, we use two set of encoding mechanisms: 1) the Record-based encoding and 2) the N-gram based encoding.

Record encoding is a general encoding method which maps every feature vector \( \vec{F} = (f_1, f_2, \ldots, f_m) \), into hyper-dimensional space. It finds the minimum and maximum feature
values and projects the range into $p$ feature levels. A set of random and orthogonal bipolar HVs $\vec{H}_l$ is assigned to every feature level. Meanwhile, for preserving the position independence of feature values in the feature vector, the Record encoding method also assigns one set of HVs to each feature values, referred to as the base HVs $\vec{H}_b$. The Record encoding method uses the level HVs to represent each feature value in the feature vector and base HVs for the position relationship of features values. Record encoding is employed by linearly combining the level HVs and base HVs:

$$\vec{H}_{\text{Record}} = \sum_{i=1}^{m} \vec{H}_{li} \ast \vec{H}_{bi}.$$  

$\vec{H}_{\text{Record}}$ is the nonbipolar encoded HV with $D$ dimensions containing integer values. Meanwhile, since base HVs are randomly generated, they are almost mutually orthogonal, which means the cosine similarity between any two base HVs $\delta(\vec{H}_{bi}, \vec{H}_{bj})$ approximately equals to 0.

We also use the N-gram encoding method, in which we employ the locality-based sparse random projection [6] as our method. If we are using the D dimensional HVs, we first extended the length of feature vector from $N$ to $D$. For instance, in MNIST dataset, when we try to encode a feature vector with $N = 768$ feature (pixel) values into $D = 10000$ dimension HVs, we first need to attach 13 duplication following the original feature vector. Meanwhile, we generate a random bipolar $D$ dimensional local-hashing HV $\vec{H}_l = \langle h_1, h_2, \ldots, h_m \rangle$. To encode the extended feature vector, N-gram encoding method deploys an N-gram sliding window and takes the dot product of the extended feature vector and projection vector in this window range: $\vec{H}_{N-\text{gram}} = \langle v_1, v_2, \ldots, v_n \rangle$. The $i$th value of $\vec{H}_{N-\text{gram}}$ equals to the dot product of the $w$ feature values from $f_i$ to $f_{i+w-1}$ and $w$ element values from $h_i$ to $h_{i+w-1}$, where $w$ is the size of sliding window.

To represent the internal data of the HDC base classifiers, we use three data widths: INT_8 and INT_16 and three different dimension settings: 1000, 5000, and 10 000 for base classifiers to enhance the diversity of base classifiers.

C. Voting Mechanism

The inference phase has two steps in the EnHDC. First, for every base classifier, we map each testing data into a query HV $\vec{H}_q$, using the same encoding method during training and calculate the cosine similarity of each class HVs with the query HV $\vec{H}_q$ in every base classifier. The inference result is pointed to the class with the highest cosine similarity. Second, we collect all the base classifier inference results in the ensemble classifier to vote the ensemble inference result.

We explored two voting mechanisms to get a better inference result and we tested two different voting strategies: 1) soft voting and 2) hard voting. Hard voting is the majority voting, while for soft voting, since each base classifier gave the inference result by cosine similarity checking, we can sum up all the related cosine distances and rank them in order, where the champion will be selected as the final result. Our test result shows that the hard voting strategy achieves better accuracy in EnHDC. Therefore, we integrate all the base inference results in ensemble classifier and use majority voting to get the ensemble inference result of the corresponding query HV.

IV. EXPERIMENTAL RESULTS

We evaluate EnHDC using four applications: 1) speech recognition (ISOLET [5]); 2) human activity recognition (HAR [1]); 3) handwritten digits (MNIST [11]); and 4) cardiocotography (CARDIO [5]).
A. Accuracy Improvement

Figs. 2 and 3 present the accuracy comparison between EnHDC configurations. The baseline has one HDC classifier while the EnHDC employs several different base classifiers.

We can observe in Fig. 2 that, EnHDC contains 8 and 16 base classifiers with different encoding methods (Record and N-gram encoding). To evaluate the performance of EnHDC, we compare three models with the dimensionality setting across \( D = 1000, 5000, 10000 \), EnHDC is showing higher accuracy than baseline models. When EnHDC has eight base classifiers, the average improvement is 3.2%. Normally HDC requires a high dimensionality such as \( D = 10000 \) to achieve satisfying performance. However, with EnHDC, we can even achieve higher accuracy with lower dimension. Across all the applications, EnHDC with eight classifiers under \( D = 10000 \) presents higher accuracy than baseline model under \( D = 10000 \). The average improvement is 1.37%.

Additionally, as shown in Fig. 3, the number of base classifiers has a notable impact on the accuracy. Without the loss of generality, our experiment features different ensemble sizes, starting from two base classifiers to 12 base classifiers. The green line shown in Fig. 3, the accuracy of EnHDC increases by adding more classifiers but comes to the vertex when using eight base classifiers for most applications. After this, the accuracy improvement is saturated. This is consistent with the ensemble theory in [2], where the performance of ensemble learning algorithms cannot constantly increase by adding an infinite amount of base classifiers. The accuracy will peak during the progress of increasing the number of classifiers, and after this peak value, the overall accuracy cannot have an obvious improvement.

The diversity enhancement can further improve the accuracy of EnHDC as shown in Fig. 3. EnHDC_enhanced classifier is the EnHDC classifier with enhanced diversity by varying encoding mechanisms (Record encoding and N-gram encoding), dimensions \((D = 1000, 5000, 10000)\), and data width \((INT_8, INT_{16})\). This figure shows that EnHDC_enhanced classifier can further improve the accuracy of EnHDC classifier by 1.2% on average across all applications, which is 4.4% improvement over baseline HDC model.

B. Model Size Reduction

Typically, HDC is required to have a high dimensionality, e.g., 10,000, to achieve a satisfying performance. For example, for CARDIO dataset in Fig. 2, the baseline HDC classifier with 10,000 dimensions has 1.9% higher accuracy than the baseline HDC classifier with 1000 dimensions. However, with ensemble learning, we can see that EnHDC with just 1000 dimensions is able to achieve similar level or even surpass the accuracy of 10,000-dimension baseline HDC classifiers across all applications. This can achieve a reduction on the HDC model size. For example, with eight base classifiers with \( D = 10000 \) dimensions, this can reduce 20% model size compared to a baseline classifier with \( D = 10000 \) dimensions.

For MNIST dataset with ten classes, we have a baseline HDC model with \( D = 10000 \) and \( INT_8 \) data width whose model size is 8 bits \( \times \) 10000 dimensions \( \times \) 10 classes \( = 800 \) Kb and EnHDC with base classifiers with \( D = 1000 \) and \( INT_8 \) whose model size is 8 bits \( \times \) 1000 dimensions \( \times \) 10 classifiers \( = 640 \) Kb. The overall model size reduction is 160 Kb for MNIST. Meanwhile, in HAR dataset with 12 classes, we have one 960-Kb baseline model with \( D = 10000 \) and EnHDC with eight 96-Kb base classifiers with \( D = 10000 \), pointing out the model size reduction for HAR is 180 Kb.

V. Conclusion

This letter presents EnHDC, an ensemble classifier designed for the emerging brain inspired HDC. EnHDC employs different base classifiers under different HV dimensions, data widths, and encoding methods. EnHDC applies the majority voting to generate the final inference result from base classifiers that are individually trained. By evaluating on four applications, we show that EnHDC can achieve higher accuracy and can reduce model size compared to baseline HDC classifiers. Additionally, by increasing the diversity of base classifiers, the classification accuracy has an enhanced improvement compared to the original EnHDC model. This letter presents effort in using an ensemble learning in HDC for boosting the performance and can enhance it for implementing in low-power platforms, such as edge computing architectures and embedded systems.

REFERENCES

[1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in Proc. Euson, vol. 3, 2013, p. 3.
[2] H. Bonab and F. Can, “Less is more: A comprehensive framework for the number of components of ensemble classifiers,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 9, pp. 2735–2745, Sep. 2019.
[3] A. Burrello, S. Benatti, K. Schindler, L. Benini, and A. Rahimi, “An ensemble of hyperdimensional classifiers: Hardware-friendly short-latency seizure detection with automatic IEEG electrode selection,” IEEE J. Biomed. Health Inform., vol. 25, no. 4, pp. 935-946, Apr. 2021.
[4] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning,” Front. Comput. Sci., vol. 14, no. 2, pp. 241–258, 2020.
[5] A. Frank. “UCI Machine Learning Repository.” 2010. [Online]. Available: http://archive.ics.uci.edu/ml
[6] M. Imani, J. Morris, J. Messerly, H. Shu, Y. Deng, and T. Rosing, “BRIC: Locality-based encoding for energy-efficient brain-inspired hyperdimensional computing,” in Proc. 56th ACM/IEEE Des. Autom. Conf. (DAC), 2019, pp. 1–6.
[7] M. Imani, D. Kong, A. Rahimi, and T. Rosing, “VoiceHD: Hyperdimensional computing for efficient speech recognition,” in Proc. IEEE Int. Conf. Rebooting Comput. (ICRC), 2017, pp. 1–8.
[8] M. Imani et al., “SearcHD: A memory-centric hyperdimensional computing with stochastic training,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 39, no. 10, pp. 2422–2433, Oct. 2020.
[9] P. Kanerva, “Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors,” Cogn. Comput., vol. 1, no. 2, pp. 139–159, 2009.
[10] G. Kanuru, M. Le Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian, “In-memory hyperdimensional computing,” Nat. Electron., vol. 3, no. 6, pp. 327–337, 2020.
[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[12] D. Ma, R. Thapa, and X. Jiao, “MoleHD: Ultra-low-cost drug discovery using hyperdimensional computing,” 2021, arXiv:2106.02894.
[13] F. R. Najafabadi, A. Rahimi, P. Kanerva, and J. M. Rabaey, “Hyperdimensional computing for text classification,” in Proc. Des. Autom. Test Eur. Conf. Exhitib. (DATE), 2016, p. 1.
[14] A. Rahimi, S. Benatti, P. Kanerva, L. Benini, and J. M. Rabaey, “Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition,” in Proc. IEEE Int. Conf. Rebooting Comput. (ICRC), 2016, pp. 1–8.