DropDim: Incorporating Efficient Uncertainty Estimation into Hyperdimensional Computing

Yang Ni[1], Hanning Chen[1], Prathyush Poduval[2], Pietro Mercati[3], and Mohsen Imani[1*]
[1]University of California Irvine, [2]University of Maryland, [3]Intel Labs
*Corresponding author: m.imani@uci.edu

***Abstract*—Research in the field of brain-inspired HyperDimensional Computing (HDC) brings orders of magnitude speedup to both Machine Learning (ML) training and inference compared to deep learning counterparts. However, current HDC algorithms generally lack uncertainty estimation On the other hand, existing solutions such as the Bayesian Neural Networks are generally slow and lead to high energy consumption. This paper proposes a hyperdimensional Bayesian framework called DropDim, which enables uncertainty estimation for the HDC-based regression algorithm. The core of our framework is a specially designed HDC encoder that maps input features to the high dimensional space with an extra layer of randomness, i.e., a small number of dimensions are randomly dropped for each input. Our key insight is that by using this encoder, DropDim implements Bayesian inference while maintaining the efficiency advantage of HDC.**

I. Introduction

In the past ten years, research in the area of deep learning observed the fast growth of Deep Neural Network (DNN) based algorithms. However, the complexity of DNNs and the computation cost of using such networks have also been increasing significantly. This inevitably leads to a surge of power consumption for training and inference

Therefore, brain-inspired computing methods such as HDC are gaining traction because of their better efficiency. In particular, HDC mimics human brain functionalities by learning in high-dimensional spaces with lightweight operations [1]. To enable HDC operations, inputs from the original low-dimensional space are encoded to vectors with thousands of dimensions, i.e., hypervectors. Recent research brings this advantage of HDC to different kinds of learning tasks, and it enables low-latency learning with less power consumption.

However, we observed that HDC-based ML algorithms still lack the ability to provide uncertainty along with regular prediction. This ability is a must for safety-critical tasks where the importance of model trustworthiness and robustness are particularly emphasized. Different from regular ML, Bayesian inference parameters have a probability distribution instead of a single value. The advantage of Bayesian statistics is that the posterior predictive distribution accounts for the noise of observation, model stochasticity, and prior knowledge about the task. Prior research works try to incorporate this advantage into the DNN learning process and propose several Bayesian Neural Networks (BNN) algorithms. Unfortunately, existing BNN algorithms bring more computations and larger energy costs in the learning, compared to already complex DNNs. We believe the lightweight HDC with uncertainty estimation is a more efficient alternative to existing BNN algorithms. We find that introducing random noise to the HDC encoding process effectively approximates the posterior distribution. This functions as the key to Bayesian inference while keeping the whole framework as lightweight as possible.

We propose DropDim, a hyperdimensional Bayesian framework that enables efficient uncertainty estimation for HDC-based regression. Our contributions includes:

- Through DropDim, we overcome a major limitation in existing HDC-based ML methods, i.e., the inability to provide uncertainty estimation. Previously, without model confidence, the usability of HDC regression algorithms is limited in safety-critical tasks.
- Our novel HDC encoder includes perturbations via randomly dropped dimensions to propagate the uncertainty estimation in our DropDim framework. It avoids complications to the original regression and simplifies the training.

II. Related Work

Bayesian Inference: There are multiple challenges in making modern ML algorithms Bayesian, especially if they are deep. Markov Chain Monte Carlo (MCMC) methods that approximate and generate samples from desired posterior distributions [2] are hardly scalable, memory-hungry, and time-consuming. Stochastic Variational Inference (SVI) learns a tractable variational distribution for the posterior but it requires significant training time and computational costs [3]. MC-Dropout [4] alleviates this overhead by leveraging neural network dropout layers but the computationally heavy DNN training process significantly increases its energy consumption.
Hyperdimensional Computing: Prior works propose HDC-based algorithms for various real-world applications [5]–[8]. These works have shown that HDC-based ML achieves notable energy savings and speedups in both training and testing, making HDC suitable for machine learning on CPUs even with tight power budgets.

III. DropDim: Enabling Efficient Bayesian HDC

Fig. 1 presents an overview of our DropDim, and compares it with the non-Bayesian hyperdimensional regression algorithm. As shown in Fig. 1(a), regular regression gives point estimates while DropDim in (b) approximates predictive distribution through a noisy HDC Bayesian encoder.

A. Hyperdimensional Regression

Vector Function Architecture (VFA) [9] defines a function space where functions can be represented using high-dimensional vectors. The VFA representation for $f(x)$ is as

978-3-9819263-7-8/DATE23/© 2023 EDAA

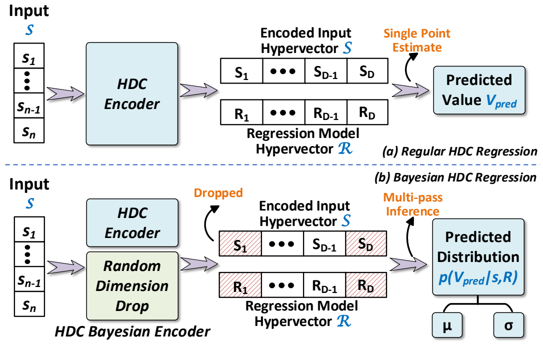

Fig. 1. Overview of DropDim

follows: $f(x) = \sum_k \alpha_k K(x - x_k) = \sum_k \alpha_k \phi(x_k)\overline{\phi(x)} = y_k^T \overline{\phi(x)}$ where y_k is the function representation and ϕ is the mapping for the kernel K. However, the exact mapping ϕ is often intractable. Work in [10] proposes that with a large but finite dimensional mapping Z, the shift-invariant kernel K can be approximated using inner products. For example, the following Z approximates the RBF kernel: $Z_D(x) = \sqrt{\frac{2}{D}} cos(\vec{H}x + \vec{B})$. $\vec{H}$ is a vector of dimension D, randomly sampled from standard Gaussian distribution, and $\vec{B}$ is sampled from the uniform distribution.

In HDC-based regression, we construct a hyperdimensional representation of the function, similar to y_k, with the mapping Z_D: $\vec{R} = \sum_k \alpha_k Z_D(x_k)$. We refer to this mapping Z_D as an HDC encoder that outputs encoded hypervectors $Z_D(x)$. The representation $\vec{R}$ shows that we can approximate the function through a weighted sum of encoded training samples. We refer to $\vec{R}$ as the model hypervector, and the inference is simply the inner product between the model and encoded hypervector. To update the model hypervector $\vec{R}$, we use prediction error as the weight for the corresponding encoded input.

B. Hyperdimensional Uncertainty Estimation

The regression mentioned above with VFA provides only point estimates with a deterministic HDC encoder, which is unable to inject uncertainty during training. We found that it is effective to randomly drop dimensions in the HDC encoder to implement stochastic perturbations. We define an encoder matrix $\mathcal{H} = \{\vec{H}_1, \vec{H}_2, \ldots, \vec{H}_n\}$ with size $n \times D$, of which the elements are randomly generated: $\vec{H}_n \in \mathcal{N}^D(0, 1)$. The bias is defined as: $\vec{B} \in \mathcal{U}^D(0, 2\pi)$. The main difference in this encoder is that some of the dimensions in the encoded output $\vec{S}$ are set to zeros or dropped. We show this modification as a randomly generated mask $\vec{M}$ with its elements $m \in Bernoulli(p_B)$.

To model the uncertainty, it is crucial to learn the posterior distribution conditioned on all training samples, which can be defined using Bayes' theorem. The intractable posterior becomes the main difficulty in calculating the accurate predictive distribution. Therefore, the only way to deal with it is by approximating the intractable posterior distribution, i.e., variational inference. A surrogate variational distribution $q(\vec{R})$ is used in place of the real posterior $p(\vec{R}|x, V)$. To ensure that $q(\vec{R})$ is a good approximation, we can minimize

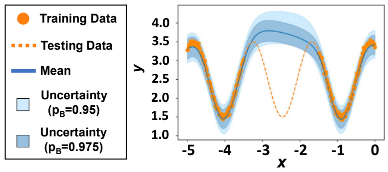

Fig. 2. DropDim uncertainty estimation with different Bernoulli probabilities

the Kullback–Leibler (KL) divergence between these two distributions. We construct a proper variational distribution $q(\vec{R})$ in DropDim by adding a random mask in the HDC Bayesian encoder. We observe that a noisy HDC encoder not only perturbs the encoded results but also can be equivalently added onto the variational distribution $q(\vec{R})$ as element dropping. KL divergence can be further approximated: $\mathcal{L}_{\mathcal{KL}} \propto \sum_k \frac{1}{2K}(V_k - \hat{V})^2 + \frac{p_B}{2\tau K}||\vec{R}||_2^2$. This can also be intuitively understood as a likelihood function plus an extra regularization term. They ensure that the regression will converge to the true values, and prevent overfitting and deviating too much from the prior distribution through the KL divergence. We learn the DropDim HDC regression model by minimizing the loss above. It maintains the advantage of regular HDC-based algorithms such as the efficient training process.

Fig. 2 shows the effect of different Bernoulli probabilities on uncertainty estimation: the range of uncertainty increases when we tune down p_B. The model is unsure about the prediction in $-3 < x < -2$ due to the lack of training data. Notice that the uncertainty is not zero even with training data because the data contains noise during training.

ACKNOWLEDGEMENTS

This work was supported in part by National Science Foundation #2127780, Semiconductor Research Corporation (SRC) AI Hardware and Hardware Security, Office of Naval Research, grants #N00014-21-1-2225 and #N00014-22-1-2067, the Air Force Office of Scientific Research under award #FA9550-22-1-0253, and a generous gift from Cisco.

REFERENCES

[1] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, vol. 1, no. 2, pp. 139–159, 2009.

[2] R. M. Neal *et al.*, "Mcmc using hamiltonian dynamics," *Handbook of markov chain monte carlo*, vol. 2, no. 11, p. 2, 2011.

[3] M. D. Hoffman *et al.*, "Stochastic variational inference," *Journal of Machine Learning Research*, 2013.

[4] Y. Gal *et al.*, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*. PMLR, 2016, pp. 1050–1059.

[5] A. Hernández-Cano *et al.*, "Reghd: Robust and efficient regression in hyper-dimensional learning system," in *DAC*. IEEE, 2021, pp. 7–12.

[6] M. Imani *et al.*, "Revisiting hyperdimensional learning for fpga and low-power architectures," in *HPCA*. IEEE, 2021, pp. 221–234.

[7] Y. Ni *et al.*, "Hdpg: hyperdimensional policy-based reinforcement learning for continuous control," in *DAC*, 2022, pp. 1141–1146.

[8] Z. Zou *et al.*, "Biohd: an efficient genome sequence search platform using hyperdimensional memorization," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022, pp. 656–669.

[9] E. P. Frady *et al.*, "Computing on functions using randomized vector representations," *arXiv preprint arXiv:2109.03429*, 2021.

[10] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in neural information processing systems*, vol. 20, 2007.