Grounded HyperSymbolic Representations Learned through Gradient-Based Optimization

Piotr Łuczak[0000–0002–2530–0283], **Krzysztof Ślot**[0000–0003–1228–0970], **Jacek Kucharski**[0000–0002–8704–1950]

Lodz University of Technology
Institute of Applied Computer Science
Stefanowskiego 18, 93-537 Łódź, Poland
pluczak@iis.p.lodz.pl

DOI:10.34658/9788366741928.51

Abstract. *Hyperdimensional computing is a novel paradigm, capable of processing complex data structures with simple operations. Its main limitations lie in the conversion of real world data onto hyperdimensional space, which due to lack of a universal translation scheme, oftentimes requires application-specific methods. This work presents a novel method for unsupervised hyperdimensional conversion of arbitrary image data. Additionally, this method is augmented by the ability of creating HyperSymbols, or class prototypes, provided that such class labels are available. The proposed method achieves promising performance on MNIST dataset, both in translating individual samples as well as producing HyperSymbols for downstream classification task.*

Keywords: *artificial intelligence, hyperdimensional computing, representation extraction, neuromorphic architectures*

1. Introduction

The problem of translating latent data representations into meaningful symbols is of paramount importance to developing real world applications of hyperdimensional computing. While capable of processing a diverse range of data types and structures in a unified manner, hyperdimensional computing [1] is currently limited by the need for custom methods of converting real world data into hyperdimensional space. Since autoencoders have attained wide-spread adoption in the domain of representation learning, they provide an ideal starting point for the construction of universal hyperdimensional encoders.

2. Related work

Hyperdimensional computing (HDC, a.k.a. Vector Symbolic Architectures), proposed by Pentti Kanerva [1] is a computational paradigm based on long (around 10^4 bits), random binary or bipolar vectors. By leveraging the properties of high-dimensional spaces (hypervectors can be thought of as vertices of a hypercube) and spreading the encoded information over all bits, this bio-inspired framework can efficiently perform complex data processing using only a few operators. Due to the simplicity of hardware implementation and resilience to noise, hyperdimensional computing has found application in domains such as event-based vision, EEG classification [2] or language classification[3]. HDC is an attractive alternative or complement to deep neural networks [4], particularly in the case of deployment to edge devices due to its ease of hardware implementation [2].

A crucial problem with hyperdimensional representation is in translating real-world data into hyperdimensional space. While a lot of classic data structures can be build with proper application of basic hyperdimensional operations, more complex data, such as images, typically require application of representation learning techniques [5]. Some successful applications have leveraged direct naive coding of values and their positions into latent vectors [6], yet the currently dominant family of methods is based on vector quantization [7]. This can be implemented e.g. by means of Sparse Distributed Memory (SDM), proposed by Pentti Kanerva [8]. As in the case of hyperdimensional vectors, stored data is distributed over the entire contents matrix $\mathbf{C}$, providing similar structural noise resilience. Both writing to ($\mathbf{C} := \mathbf{C} + ((\mathbf{A}\mathbf{x}_{addr}) \geq d)\mathbf{x}_{data}^T$) and reading from ($\mathbf{x}_{data} = sign(\mathbf{C}^T((\mathbf{A}\mathbf{x}_{addr}) \geq d))$) the SDM are based on the similarity between $\mathbf{x}_{addr}$ and the address matrix $\mathbf{A}$.

3. Methods

The main contribution of this work is a novel method for deriving hyperdimensional data representation, based on a modified Autoencoder than can be trained using standard gradient-based methods, as shown in Fig.1. An ancillary contribution of this work is a proposal of the simplified, bidirectional version of SDM.

The mapping from latent space to hypervectors is simply a linear composition of seed hypervectors stored in the value book $\mathbf{V} \in \mathbb{B}^{n \times v}$, with weights based on the similarity of a latent vector $\mathbf{k} \in \mathbb{Z}^{k \times 1}$ to learned values in the key book $\mathbf{K} \in \mathbb{Z}^{n \times k}$:

$$\mathbf{v} = \tanh(\mathbf{V}^T(\mathbf{k} \approxeq \mathbf{K})) \tag{1}$$

The inverse mapping, that is moving from the hyperdimensional space to Autoencoder's latent space, follows the same principle. In this case the output is a linear composition of latent vectors from the key book based on the similarity between hypervector $\mathbf{v} \in \mathbb{B}^{v\times 1}$ and the value book:

$$\mathbf{k} = \mathbf{K}^T softmax(\mathbf{V}\mathbf{v}) \tag{2}$$

The complete structure, showing the data flow through the proposed model is shown in Fig.1.

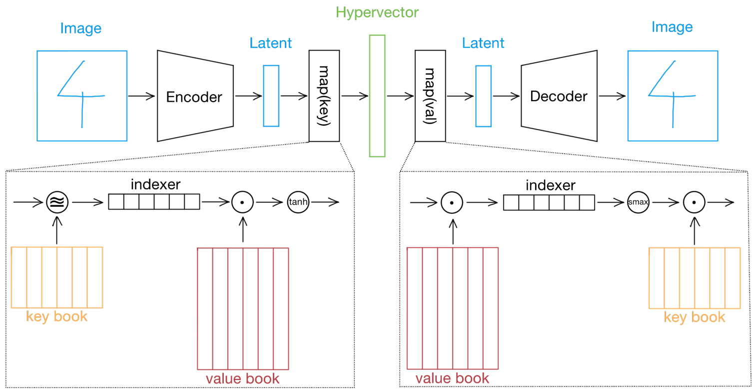


Figure 1: Structure of the hyperdimensional autoencoder. Source: own work.

The final step of deriving the HyperSymbolic representation is grounding it, that is building hyperdimensional prototypes associated with each known label. The construction of a class prototype can be done by simply bundling hypervector representations derived for a set of training images with the same label. The number of components can be relatively low, oftentimes requiring only around 100 examples from each class.

While hypervectors can be compared using the dot product, a gradient-friendly method of measuring floating point vector similarity was needed to enable bidirectional mapping between floating point and hyperdimensional vectors. For this purpose, the relaxed equality operator (equation 3) [9] has been chosen and extended into vector version as the mean of elementwise computations.

$$a \approxeq b \equiv sech^2\left(\frac{b-a}{2\beta}\right) \tag{3}$$

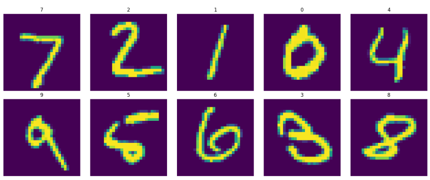
The support of this operator can be controlled by changing the β parameter. Given that the input to the hyperdimensional encoder has been produced by the sigmoid layer of the convolutional encoder, the value of β was tuned so that the furthest 20% of values were not matched.

4. Results

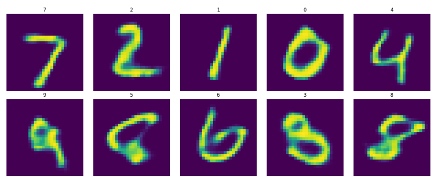
The proposed model has been evaluated on the MNIST dataset after being trained for 100 epochs with the PyTorch's implementation of the Adam optimiser.

Since the model had been designed to behave like a classic autoencoder, it was trained using mean square error loss. Data augmentation, learning rate schedulers and pretraining were omitted deliberately, in order to evaluate the learning ability of the hyperdimensional component without additional confounding factors.

The performance of the proposed model can be evaluated in a twofold manner. Firstly, the model can be assessed simply as an autoencoder, as shown in figure 2. The set of images in Fig. 2a has been randomly sampled from the testing set, while the set of images in Fig 2b is the output of the autoencoder for those images. As it can be observed, the input handwriting exhibits a number of deviations from the "ideal" digit shapes, however the autoencoder is capable of providing their legible reconstructions.



(a) Random sample of input data.



(b) Autoencoder output.

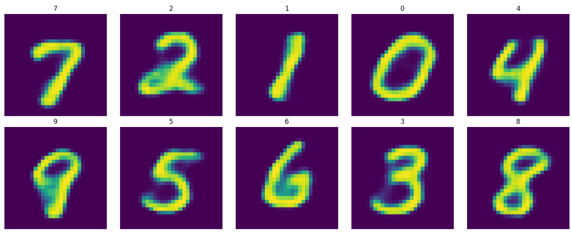
Figure 2: Autoencoder performance on a random sample of dataset. Source: own work.

Secondly, the model can be evaluated based on its ability to assemble well formed HyperSymbols, as shown in Fig.3. While the training dataset contained digits with varying degrees of nonideality, the prototypes constructed from random samples of 150 representatives of each digit, strongly resemble the "textbook" versions of digits.This indicates that the prototypes are indeed well formed, as the model was capable of extracting the "ideal" shapes of digits, from their non-ideal representatives. The result of decoding of these prototypes can be seen in fig. 3a.

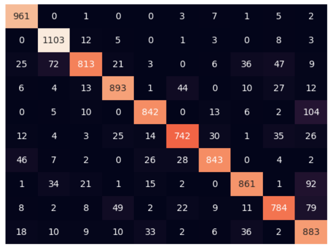
HyperSymbols can be used for a number of downstream applications, such as encoding expert knowledge. For the derived HyperSymbols, we test their generic nature using a task of image classification. As shown in Fig.3b, the hyperdimensional classifier achieves promising performance, with the average accuracy of 0.8725. While the classification performance is not perfect, the hyperdimensional component of the model was trained with only 150 occurrences of each digit, which is significantly less than required by most neural networks.

5. Conclusion

The proposed model enables simple conversion of real data into hyperdimensional space and vice versa, with simple, unsupervised gradient-based training. It enables easy inclusion of HDC paradigm in real world applications, without incur-



(a) Decoded hyperdimensional prototypes for all classes in MNIST dataset.



(b) Confusion matrix for hyperdimensional classification.

Figure 3: Performance of the hyperdimensional classifier. Source: own work.

ring the cost of developing custom, application-specific encoding schemes. Future work on this method could investigate the degree of transferability of trained encoders between similar domains, as well as the impact of state-of-the-art neural network training techniques and models on the quality of obtained HyperSymbols.

Acknowledgements

This work was funded by European Union's Horizon 2020 research and innovation programme under grant agreement no 101016734. This work has been completed while the 1st author was the Doctoral Candidate in the Interdisciplinary Doctoral School at the Lodz University of Technology, Poland.

References

[1] Kanerva P., *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*, *Cognitive Computation*, 2009, vol. 1, no 2, doi: 10.1007/s12559-009-9009-8.

[2] Rahimi A., Kanerva P., Benini L., Rabaey J.M., *Efficient Biosignal Processing Using Hyperdimensional Computing: Network Templates for Combined Learning and Classification of ExG Signals*, *Proceedings of the IEEE*, 2019, vol. 107, no 1, ISSN 1558-2256, doi: 10.1109/JPROC.2018.2871163.

[3] Joshi A., Halseth J.T., Kanerva P., *Language Geometry Using Random Indexing*, [In:] *Quantum Interaction*, Lecture Notes in Computer Science, Springer International Publishing, doi: 10.1007/978-3-319-52289-0_21.

[4] Łuczak P., Ślot K., Kucharski J., *Combining Deep Convolutional Feature Extraction with Hyperdimensional Computing for Visual Object Recognition*,

[In:] *2022 International Joint Conference on Neural Networks (IJCNN)*, ISSN 2161-4407, doi: 10.1109/IJCNN55064.2022.9892281.

[5] van den Oord A., Vinyals O., kavukcuoglu k., *Neural Discrete Representation Learning*, [In:] *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc.

[6] Nazemi M., Fayyazi A., Esmaili A., Pedram M., *SynergicLearning: Neural network-based feature extraction for highly-accurate hyperdimensional learning*, [In:] *Proceedings of the 39th International Conference on Computer-Aided Design*, ICCAD '20, ACM, doi: 10.1145/3400302.3415696.

[7] Mitrokhin A., Sutor P., Summers-Stay D., Fermüller C., Aloimonos Y., *Symbolic Representation and Learning With Hyperdimensional Computing*, *Frontiers in Robotics and AI*, 2020, vol. 7, ISSN 2296-9144.

[8] Kanerva P., *Sparse Distributed Memory and Related Models*, [In:] *Associative Neural Memories: Theory and Implementation*, Oxford University Press, ISBN 978-0-19-507682-0, 1993, pp. 50–76.

[9] Petersen F., *Learning with Differentiable Algorithms*, 2022, doi: 10.48550/arXiv.2209.00616.