Comparing Hyperdimensional Computing to Deep Learning for Natural Language Processing Tasks

Todd Morrill
Computer Science Department
Columbia University
tm3229@columbia.edu

Satyam Sharma
Computer Science Department
Columbia University
ss6522@columbia.edu

***Abstract*—In this project, we will compare the performance of deep learning models (e.g. Transformers, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) models) to HDC models on a variety of NLP tasks using a range of metrics and evaluate their relative strengths and weaknesses.**

***Index Terms*—hyperdimensional computing, HDC, deep learning, natural language processing, NLP**

I. Introduction

Deep learning is currently the dominant approach in natural language processing (NLP). In particular, the Transformer [?] architecture has been shown to be effective for a wide range of NLP tasks, including language modeling (i.e. next word prediction), document retrieval, and document classification [?]. However, deep learning models require a large amount of training data and are often memory and energy intensive, which limit their usability on low-resource devices (e.g. smartphones). Hyperdimensional computing (HDC), on the other hand, is a neuro-inspired approach to machine learning that is memory and energy efficient and may require far less training data to achieve suitable levels of accuracy [?]. In short, HDC typically represents data as random high dimensional vectors (e.g. a word may be represented in $\{-1, 1\}^{10,000}$). HDC uses a variety of elementwise operations to operate on this data. In particular, addition is coordinatewise majority, mulitiplication is coordinatewise XOR, permutation is a rotation of coordinates (i.e. shift to the right), and comparison can be done using a hamming distance or cosine distance. These operations allow HDC to perform next word prediction, document retrieval, and classification, among other tasks.

In this project, we will compare the performance of deep learning models (e.g. Transformers, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) models) to HDC models on a variety of NLP tasks using a range of metrics and evaluate their relative strengths and weaknesses.

II. Models and Data Set(s) Description

We aim to compare the performance of deep learning and HDC on the following NLP tasks:

1) missing word prediction - predicting the masked word in a given sentence
2) document retrieval - querying semantically similar content across several documents
3) language identification - detecting the language of a given sentence.

To determine relative strengths and weaknesses, we will be capturing the following metrics:

1) accuracy relative to different dataset sizes
2) training and inference time
3) energy consumption as measured by FLOPs [?]
4) robustness against input data corruption

III. Training and Profiling Methodology

We will primarily use Python for our implementation and use PyTorch and Hugging Face [?] to implement the deep learning models and TorchHD [?] to implement the HDC models. We intend to implement and run our experiments on a combination of laptops, Habanero, Google Colab, and a personal deep learning server. We plan to train and evaluate on the following corpora:

1) missing word prediction - Wikipedia [?]
2) document retrieval - the BEIR benchmark [?]
3) language identification - the Wortschatz Corpora [?].

IV. Performance Tuning Methodology

V. Experimental Results

VI. Conclusion

VII. Appendix

TABLE I
HDC accuracy scores by dataset size.

Examples	Dataset Pct.	Accuracy
21	0.0001	0.2682
210	0.0010	0.8658
2100	0.0100	0.9590
4200	0.0200	0.9664
10501	0.0500	0.9697
21003	0.1000	0.9736
42006	0.2000	0.9727
105016	0.5000	0.9730
210032	1.0000	0.9740

TABLE II
DEEP LEARNING ACCURACY SCORES BY DATASET SIZE.

Examples	Dataset Pct.	Accuracy
21	0.0001	0.0476
210	0.0010	0.0541
2100	0.0100	0.7886
4200	0.0200	0.8629
10501	0.0500	0.9513
21003	0.1000	0.9652
42006	0.2000	0.9793
105016	0.5000	0.9855
210032	1.0000	0.9898

TABLE III
DEEP LEARNING SPEED ANALYSIS.

Model	Training-Time	Testing-Time
HDC	101.395437	10.215967
HDC-Optimized	44.142246	4.526111
distilbert-base-uncased	1559.166500	88.025828

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[2] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," *CoRR*, vol. abs/1905.00537, 2019. [Online]. Available: http://arxiv.org/abs/1905.00537

[3] A. Rahimi, P. Kanerva, and J. M. Rabaey, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing." Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2934583.2934624

[4] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Compute and energy consumption trends in deep learning inference," 2021.

[5] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *CoRR*, vol. abs/1910.03771, 2019. [Online]. Available: http://arxiv.org/abs/1910.03771

[6] M. Heddes, I. Nunes, P. Vergés, D. Desai, T. Givargis, and A. Nicolau, "Torchhd: An open-source python library to support hyperdimensional computing research," 2022.

[7] Nov 2022. [Online]. Available: https://huggingface.co/datasets/wikipedia

[8] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models," *CoRR*, vol. abs/2104.08663, 2021. [Online]. Available: https://arxiv.org/abs/2104.08663

[9] U. Quasthoff, M. Richter, and C. Biemann, "Corpus portal for search in monolingual corpora," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA), May 2006. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2006/pdf/641$_{p}df.pdf$

TABLE IV
DEEP LEARNING FLOP ANALYSIS.

Model	Parameters	FLOPs
HDC	490,000	210,000
distilbert-base-uncased	43,135,509	936,371,712