Comparing Hyperdimensional Computing to Deep Learning for Natural Language Processing Tasks

HPML Final Project Proposal

Todd Morrill
tm3229@columbia.edu

Satyam Sharma
ss6522@columbia.edu

Friday, March 24, 2023

1 Goals and Objectives

Deep learning is currently the dominant approach in natural language processing (NLP). In particular, the Transformer [8] 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 [9]. 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 [6]. 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.

2 Challenges

HDC is not as established as deep learning, and as a result, tools and frameworks may not be complete or have popular community support. The lack of support may create challenges for us in implementing HDC systems. In particular, we potentially have to develop some basic building blocks from scratch, e.g., implementing vectorized operations on the GPU for HDC algorithms.

3 Approach and Techniques

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 [3]
4. robustness against input data corruption

Our primary focus will be on the Transformer architecture but if time permits, we may also examine the performance of LSTMs and CNNs.

4 Implementation Details

We will primarily use Python for our implementation and use PyTorch and Hugging Face [10] to implement the deep learning models and TorchHD [4] 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 [1]
2. document retrieval - the BEIR benchmark [7]
3. language identification - the Wortschatz Corpora [5].

5 Demo Planned

Besides doing the presentation in the class, we intend to build a web application that allows user to also test the aforementioned techniques with text input. We intend to build this using Streamlit[1] or Gradio[2], which will interface with our trained models.

[1] `https://github.com/streamlit/streamlit`

References

[1] Nov. 2022. URL: https://huggingface.co/datasets/wikipedia.

[2] Abubakar Abid et al. "Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild". In: *arXiv preprint arXiv:1906.02569* (2019).

[3] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. *Compute and Energy Consumption Trends in Deep Learning Inference*. 2021. arXiv: 2109.05472 [cs.LG].

[4] Mike Heddes et al. *Torchhd: An Open-Source Python Library to Support Hyperdimensional Computing Research*. 2022. arXiv: 2205.09208 [cs.LG].

[5] Uwe Quasthoff, Matthias Richter, and Christian 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. URL: http://www.lrec-conf.org/proceedings/lrec2006/pdf/641_pdf.pdf.

[6] Abbas Rahimi, Pentti Kanerva, and Jan M. Rabaey. "A Robust and Energy-Efficient Classifier Using Brain-Inspired Hyperdimensional Computing". In: Association for Computing Machinery, 2016. ISBN: 9781450341851. DOI: 10.1145/2934583.2934624. URL: https://doi.org/10.1145/2934583.2934624.

[7] Nandan Thakur et al. "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models". In: *CoRR* abs/2104.08663 (2021). arXiv: 2104.08663. URL: https://arxiv.org/abs/2104.08663.

[8] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[9] Alex Wang et al. "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems". In: *CoRR* abs/1905.00537 (2019). arXiv: 1905.00537. URL: http://arxiv.org/abs/1905.00537.

[10] Thomas Wolf et al. "HuggingFace's Transformers: State-of-the-art Natural Language Processing". In: *CoRR* abs/1910.03771 (2019). arXiv: 1910.03771. URL: http://arxiv.org/abs/1910.03771.