Hybrid Neural Network Training on Diverse Hardware

Fabian Arun Panaite
University of Petrosani, Romania
Monica Leba
University of Petrosani, Romania

Abstract

This study presents a comprehensive comparison of neural network training hardware structures, focusing on the performance of CPU and GPU processors under varying data storage conditions (SSD drive and RAM disk). Initially, the training efficiency and speed of neural networks are analysed using a CPU processor, with data stored first on an SSD drive and subsequently in a RAM disk to evaluate the impact of data retrieval speeds on training times and accuracy. The analysis is then extended to GPU processors, renowned for their superior parallel processing capabilities, under identical data storage conditions to discern the benefits and limitations of each hardware setup in neural network training scenarios. Additionally, a novel hybrid architecture is proposed, combining either CPU or GPU processors with the concept of memory sprites—a technique borrowed from the age of video game development for optimizing graphics on less capable hardware. This approach aims to leverage the advantages of both processing units while mitigating their weaknesses, offering a potentially superior solution for training complex neural networks efficiently on diverse hardware platforms.

Keywords: Neural Networks, CPU, GPU, SSD, RAM, memory sprites
JEL classification: L63, L86

Paper type: Research article
Received: 5 February 2024
Accepted: 28 May 2024

DOI: 10.54820/entrenova-2024-0023

Introduction

An ingenious construction within our computers that uses a block of random access memory (RAM) as if it were a disk drive. Often referred to as a virtual disk, it is a marvel of modern computing, using software to emulate a hard drive but operating at a pace that leaves traditional storage methods in the dust.

Speed

A RAM disk operates with a swiftness akin to lightning compared to the steady plod of traditional hard drives or even the brisk trot of solid-state drives. It uses the computer's main memory, a realm where read and write speeds are unparalleled.

Volatility

Like a brilliant flash of light that fades as quickly as it appears, the data on a RAM disk is ephemeral, vanishing when the computer loses power or is restarted. This transient nature renders the RAM disk unsuitable for storing treasures of information over long periods, yet it is unmatched for temporary storage needs where speed is of the essence.

Usage Scenarios

RAM disks find their place in environments that thirst for rapid data access and can forego long-term retention. They shine in system caches, development environments, high-performance servers, and scenarios where data voluminously flows but only needs brief harbouring.

Size Limitations

The expanse of a RAM disk is tethered to the amount of RAM at one's disposal. Allocating RAM to this disk subtracts from the quantum available for other tasks, thus a judicious balance must be struck based on the system's demands.

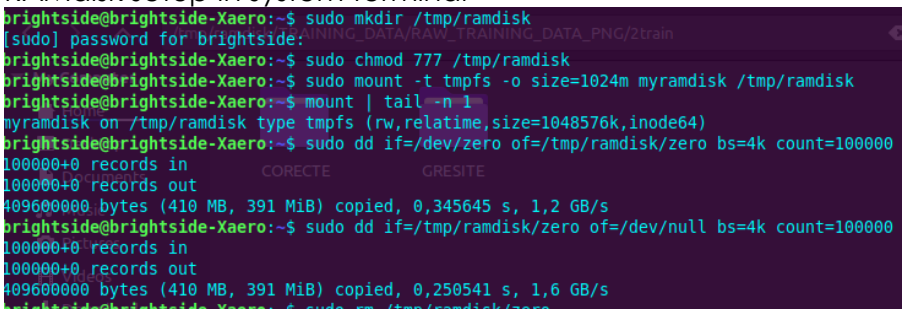
Overall, the RAM disk, with its exceptional velocity, stands as a paragon of performance for specific applications where persistence of data takes a back seat to the need for speed. Herein lies a tool of magnificent utility, worthy of consideration by those who prioritize efficiency in their digital endeavours.

Figure 1a

System properties

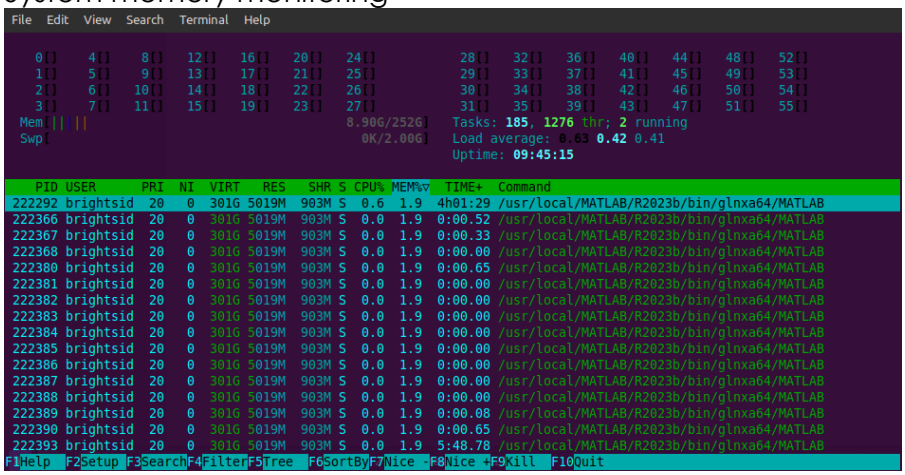
Source: Author's illustration

Figure 1b
RAMdisk setup in system terminal



Source: Author's illustration

Figure 1c
System memory monitoring



Source: Author's illustration

Literature review

The research (Bahn and Cho, 2020) evaluates the re-evaluation of memory hierarchies due to advancements in non-volatile memory (NVM) technologies like PCM and STT-MRAM, focusing on performance implications and design considerations for system software development in evolving storage landscapes. It identifies pros such as performance optimization and reduced power consumption, alongside challenges like prefetching ineffectiveness and the need for software adaptations (Đorđević et a., 2006). Study (Đorđević et a., 2006) investigates NVRAM-based journaling in Linux, analyzing impacts on filesystem performance and resource efficiency. It outlines pros like novel approach and practical implications but notes cons like simulation errors and limited scope. Discoveries include minimal and optimal NVRAM sizes for effective journaling and the performance impact across different journaling modes. Research delves into extreme-scale HPC systems, proposing a new architecture with node-local persistent storage to address efficiency, scalability, and reliability challenges at exascale levels. Pros include improved performance and scalability, with cons like complexity and limited evaluation. The study emphasizes innovative storage architecture and integration of IDA and erasure coding (Diao et al., 2012). The study (Diao et al., 2012) on implementing a RAM disk through Windows drivers highlights the high-speed storage solution provided by RAM disks, with pros including reliability and customizable size, but notes cons like limited storage capacity and volatile storage. Key discoveries revolve around the successful implementation of a resizable RAM disk using Windows Driver Foundation (WDF) (Najeeb, 2017). Research (Najeeb, 2017) on

image compression using MLP and Back-Propagation algorithm addresses efficient compression techniques, balancing compression ratio with image quality. Pros include efficiency and adaptability, but challenges like complexity and resource intensity are noted. The study demonstrates the effectiveness of MLP for high-quality image compression (Lutnick et al., 2019). The research (Lutnick et al., 2019) applies CNNs to biological datasets, especially in medical imaging, overcoming the lack of centrally curated training sets with the Human AI Loop (H-AI-L) method. It identifies pros like innovative approach and clinical relevance but faces challenges in data annotation and resource intensity. The H-AI-L technique shows promise in segmenting tissue and improving the accuracy of expert annotations (Serra et al., 2020). Study (Serra et al., 2020) on lossless compression of deep neural networks explores reducing network size without losing expressiveness using MILP and ℓ1 regularization. Despite its complexity and limited scope to specific types of networks, it provides a potential path for efficient deployment of complex models on resource-constrained devices (Alirezazadeh et al., 2022). Research (Alirezazadeh et al., 2022) compares plant disease classification using deep learning with attention mechanisms, highlighting improved performance through CBAM. Pros include efficiency and real-world application, but the approach has limitations in model selection and generalization. Key discoveries show CBAM's potential in enhancing feature focus and classification accuracy in limited data scenarios (Ramshankar, 2023). The research (Ramshankar, 2023) on protecting DNN training data on edge devices using TEEs discusses the partitioning of DNNs to safeguard privacy, emphasizing compatibility with Python-based ML frameworks and efficient resource usage. However, it notes performance impacts due to TEEs' limitations and the complexity of implementation (Truong et al., 2021). Study (Truong et al., 2021) focuses on addressing performance bottlenecks in TEEs for deep learning inference, proposing techniques like y-plane partitioning and weight quantization/compression. It notes innovative solutions and efficiency improvements but mentions limited model scope and hardware dependency (Babar et al., 2023). Research (Babar et al., 2023) surveys secure execution of neural networks in TEEs, providing a comprehensive review of techniques and identifying adversarial threats. It underscores the need for further research on resource optimization and side-channel vulnerability mitigation (Olivier and Raj, 2022). Study (Olivier and Raj, 2022) examines the adversarial robustness of the Whisper ASR model, revealing vulnerabilities to adversarial perturbations despite its robustness to noise. The research suggests future directions in developing more robust ASR models and extending evaluations to other architectures (Navarro et al., 2020). Research (Navarro et al., 2020) leverages GPU tensor cores for accelerating non-Deep Learning computations, achieving significant performance gains through a novel reduction algorithm. The approach shows potential for extending to other computational patterns, with emphasis on improving scalability and precision (Shacklett et al., 2021). The study (Shacklett et al., 2021) enhances reinforcement learning training in 3D environments through batch simulation, achieving high throughput and maintaining sample efficiency. It proposes exploring more complex environments and integrating other AI techniques for improved agent performance (Athil et al., 2014). Research (Athil et al., 2014) optimizes matrix multiplication on GPUs using CUDA techniques, focusing on efficiency improvements in scientific and commercial applications. Future directions include developing more advanced memory management techniques and exploring dynamic thread scheduling for enhanced performance (Ostrowski, 2020). Overall, these studies address a wide range of topics from memory technologies and image compression to neural network security and computational efficiency, each

contributing valuable insights and outlining areas for future research to further advance technology and application effectiveness.

Methodology

During the experiments and tests, it has been discovered that the transfer of MATLAB files to a RAM disk does not markedly enhance the training speed. Computation Intensive: The heart of neural network training thrums with computational demands. The real challenge lies in the raw power of the CPU or GPU, which are the workhorses of this process. These are the arenas where battles for speed are truly won or lost, far removed from the mere retrieval of program files. Data Access: In the dance of training a neural network, the most significant movements are those involving the training data itself. The efficiency with which this data pirouettes into memory is paramount. Techniques within MATLAB, such as datastores or specialized data handling methods, are the choreographers ensuring the performance flows smoothly, not the location of the program files. Disk I/O: Once the curtain rises and MATLAB begins its performance, the program files are generally summoned into memory a single time. Minimal disk interactions occur thereafter, meaning the stage is already set whether or not the files reside on a RAM disk. Caching: Modern operating systems, much like wise stage managers, are adept at keeping the most frequently used scripts—our MATLAB files—ready at hand in the wings (RAM), thereby diminishing the practical benefits of manually moving them to a RAM disk. Temporary Files and Workspace: The spotlight might more effectively shine on MATLAB's temporary directories or workspaces, where the hustle of reading and writing intermediate data is relentless. Here, a RAM disk could play a more critical role, although this too is a mere piece of the puzzle compared to the grand ensemble of CPU and GPU capabilities. In summary, while the idea of placing MATLAB program files on a RAM disk is akin to fine-tuning an instrument, the neural network training demands a more robust orchestration of computational resources, data handling, and system architecture. This is where the true performance lies, not merely in the speed at which the files can be accessed.

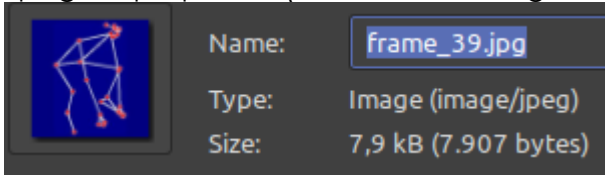
In the quest to speed up neural network training in MATLAB, it is unyielding to consider that, a machine with faster CPU or GPU, GPU acceleration in MATLAB, neural network structure optimization and data format. The image format proven to be the best fit for neural network training is rather a choice bent on the destination task that the neural network would be set up for. A discerning look at the strengths and considerations of each image format:

JPEG - Highpoint: The art of JPEG lies in its compression, allowing it to occupy less space on the RAM disk. This compression can enable a greater number of images to be stored, potentially accelerating the data loading sequence.

Consideration: However, JPEG employs lossy compression, which may erode image quality, a compromise that could be significant in training where every detail counts.

Figure 2a

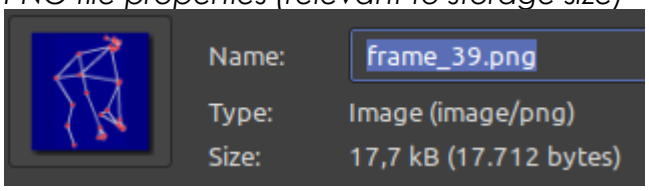
Jpeg file properties (relevant to storage size)



Source: Author's illustration

PNG - Highpoint: The integrity of PNG is in its lossless compression, safeguarding the quality of each image. This characteristic is paramount for training tasks where precision in image detail directly influences the performance of the neural network. Consideration: The flip side of PNG is its larger file size compared to JPEG, which might constrain the number of images storable on a RAM disk concurrently.

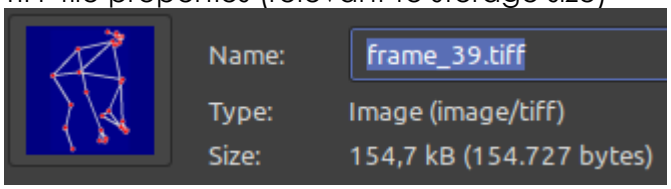
Figure 2b
PNG file properties (relevant to storage size)



Source: Author's illustration

TIFF - Highpoint: TIFF excels with its support for lossless compression and higher bit depths, offering a robust format for high-quality image data essential for precise neural network training. Consideration: The substantial size of TIFF files can hamper their efficiency in terms of storage and speed in a RAM disk setup.

Figure 2c
TIFF file properties (relevant to storage size)



Source: Author's illustration

For preserving quality, PNG or TIFF are best fits for their unfaltering commitment to lossless compression, while for maximizing space efficiency, JPEG to accommodate a higher volume of images, accepting some quality loss for greater storage efficiency. The decision ultimately rests on the precise needs of the neural network training endeavour, balancing the scales between maintaining image fidelity and optimizing the use of RAM disk space. This is a classic example of the nuanced decisions that must be made in the pursuit of technological excellence. For this specific task set, the data had been chosen to be in TIFF format.

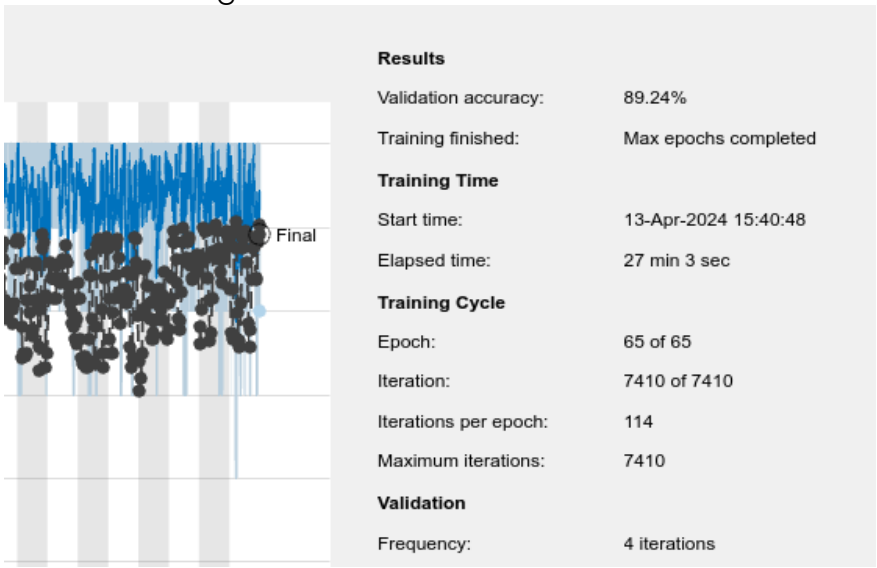
Results and discussion

Training duration differences are small, neural networks trained with data stored on a RAMdisk show considerably higher validation accuracy compared to those trained using data stored on an HDD or SSD. Several factors potentially contribute to this discrepancy.

Data Access Speed and Throughput: The most apparent difference between using a RAMdisk and using traditional storage like HDDs or SSDs is the speed at which data can be accessed. RAMdisk is significantly faster, which means data can be loaded into the model more quickly. This is particularly advantageous in scenarios where data needs to be fetched and pre-processed on-the-fly during training. However, if training durations are the same, it suggests that data loading times might not be the bottleneck in this case. One potential issue with HDDs and SSDs, especially in large-scale data environments, could be errors in data reading or subtle corruption that

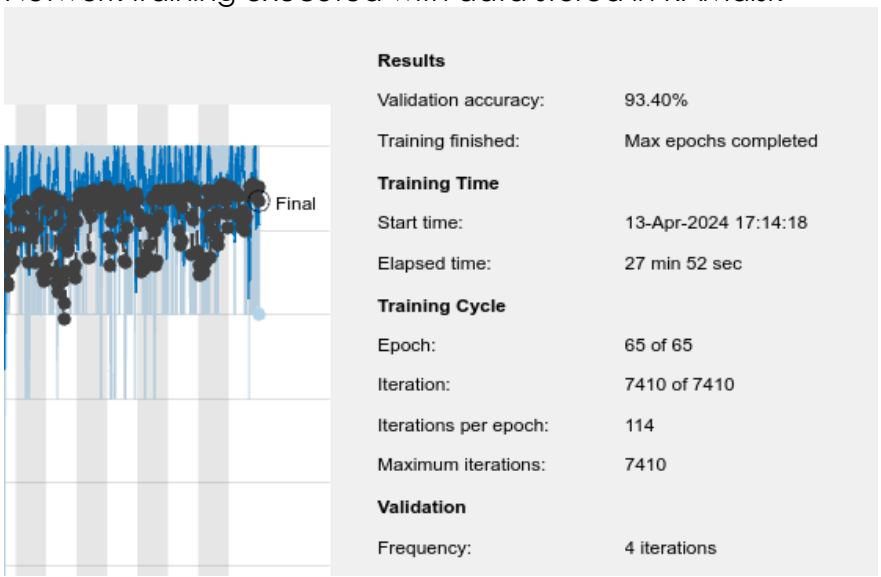
doesn't halt the process but affects data quality. RAMdisks, generally being more reliable in terms of data fetching, might avoid these issues, resulting in cleaner or more consistent data during training. Training processes might use temporary storage differently based on the medium. For example, certain computations might be cached more efficiently when using RAMdisk due to its high speed, leading to more accurate or stable training iterations. The setup for training with RAMdisk vs. HDD/SSD involves some differences in how data is handled, pre-processed, or batched. These differences inadvertently impact how well the model learns patterns from the data. Differences in the underlying hardware and software configurations when switching storage mediums could introduce variables that affect training. For instance, using RAMdisk might allow for different configurations or optimizations in the training software that aren't feasible with slower storage solutions.

Figure 3a
Network training executed with data stored on SSD



Source: Author's illustration

Figure 3b
Network training executed with data stored in RAMdisk



Source: Author's illustration

Conclusion

Training neural networks had been proven to have not necessarily faster, but actually more accurate results in the training process. Neural networks trained on data stored on a RAMdisk exhibit significantly higher validation accuracy compared to those using HDD or SSD storage, despite similar training durations. The primary advantage of RAMdisk lies in its faster data access speed, enabling quicker data loading and processing during training. This speed can be crucial when data needs on-the-fly fetching and preprocessing. HDDs and SSDs may encounter data reading errors or subtle corruption, affecting data quality. RAMdisk's reliability ensures cleaner data, potentially leading to more consistent training outcomes. Differences in hardware and software setups when using RAMdisk versus traditional storage may also influence model performance due to variations in data handling, preprocessing, and training software configurations.

References

1. Alirezazadeh, P., Schirrmann, M., & Stolzenburg, F. (2022). Improving deep learning-based plant disease classification with attention mechanism. *Journal/Institution Details*.
2. Athil, T., Christian, R., & Reddy, Y. B. (2014). CUDA memory techniques for matrix multiplication on Quadro 4000. *11th International Conference on Information Technology: New Generations*. *Grambling State University*.
3. Babar, M. F., & Hasan, M. (2023). Trusted deep neural execution—A survey. *Electrical Engineering and Computer Science, Washington State University*.
4. Bahn, H., & Cho, K. (2020). Implications of NVM based storage on memory subsystem management. *Department of Computer Engineering, Ewha University*. https://doi.org/10.3390/xxx123
5. Diao, J., Guo, J., Yu, H., Sun, Z., Liu, H., & Nie, H. (2012). The implementation of a RAM disk. *School of Electronic Science and Engineering, National University of Defense Technology*.
6. Đorđević, B., Obradović, S., Maček, N., & Mišković, S. (2006). NVRAM-based journaling in Linux. *Advanced School of Electrical Engineering, Belgrade*.
7. Lutnick, B., Ginley, B., Govind, D., McGarry, S. D., LaViolette, P. S., Yacoub, R., Jain, S., Tomaszewski, J. E., Jen, K.-Y., & Sarder, P. (2019). An integrated iterative annotation technique for easing neural network training in medical image analysis. *Institution Details*.
8. Najeeb, H. D. (2017). Artificial neural network for TIFF image compression. *Dept. of Public Relations, College of Media, University of Al Iraqia*.
9. Navarro, A., Carrasco, R., Barrientos, R. J., Riquelme, J. A., & Vega, R. (2020). GPU tensor cores for fast arithmetic reductions. *Institution Details*.
10. Olivier, R., & Raj, B. (2022). There is more than one kind of robustness: Fooling Whisper with adversarial examples. *Carnegie Mellon University*.
11. Ostrowski, E. (2020). Survey of alternative hardware for neural network computation in the context of computer vision. *University of Cologne*.
12. Ramshankar, G. (2023). Protect privacy of training data in deep neural networks on edge using trusted execution environments. *Preprint*.
13. Serra, T., Kumar, A., & Ramalingam, S. (2020). Lossless compression of deep neural networks. *Bucknell University & The University of Utah*. https://arxiv.org/abs/2001.00218v3
14. Shacklett, B., Wijmans, E., Petrenko, A., Savva, M., Batra, D., Koltun, V., & Fatahalian, K. (2021). Large batch simulation for deep reinforcement learning. *Stanford

University, Georgia Institute of Technology, Intel Labs, University of Southern California, Simon Fraser University*. https://arxiv.org/abs/2103.07013v1
15. Truong, J.-B., Gallagher, W., Guo, T., & Walls, R. J. (2021). Memory-efficient deep learning inference in trusted execution environments. *Worcester Polytechnic Institute*. https://arxiv.org/abs/2104.15109v2

About the authors

Fabian Arun Panaite, Ph.D. student Eng. and assistant at University of Petrosani, Romania with a thesis that approaches methods for human posture recognition. He graduated System Control Engineering specialization both bachelor and master at the University of Petrosani, Romania. His main research interests are operating systems and multimedia systems. The author can be contacted at e-mail: **fabianpanaite@upet.ro**

Prof. Monica Leba is a Ph.D. supervisor in field of System Control Engineering in Computer and System Control Engineering Department at University of Petrosani, Romania. She has a PhD in System Control Engineering from University of Petrosani, a bachelor's degree in applied informatics and Master's Degree in Computer Engineering from University of Petrosani. She is a member of IEEE and IFAC, Computers for Control Technical Committee. Her research interests are in applied informatics, modelling-simulation, algorithms design. She has more than 100 refereed research articles in international journals, conferences and international patents prizes. The author can be contacted at e-mail: **monicaleba@upet.ro**