Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

UDC - 004.725:004.852

RECOMMENDATIONS FOR THE DEPLOYMENT OF EQUIPMENT FOR TRAINING NEURAL NETWORKS

Timur V. Jamgharyan
National Polytechnical University of Armenia
105 Teryan St, 0009, Yerevan
t.jamharyan@politechnic.am
ORCID iD: 0000-0002-9661-1468
Republic of Armenia

https://doi.org/10.56243/18294898-2024.3-84

Abstract

The paper presents recommendations for choosing a hardware resource when training neural networks. Analytical expressions are considered for calculating the computational complexity when deploying direct propagation neural networks of various dimensions. Some problems arising in the selection and configuration of hardware on which neural networks are trained are formulated. An assessment of the use of libraries/frameworks of various neural networks is given.

A set of recommendations for the mutual arrangement of hardware modules is developed.

Keywords: hyperparameter, machine learning model, training set, logical transparency, TensorFlow, Translation lookaside buffer, ParaDnn.

Introduction

Deploying systems using Machine Learning (ML), an important place is occupied by the preparation and calculation of the necessary hardware. To obtain the most effective result, it is important to correctly arrange the various components of the hardware resource, as well as to evaluate the system compatibility of the hardware and software component. Hardware compatibility with specialized libraries allows to speed up modeling and training processes of ML. Depending on the programming language used, the set of ML libraries also changes *(TensorFlow, Scikit-learn, Keras, NumPy, Matplotlib, NLTK*[1]*, Gensim, Armadillo, TensorFlow, Tiny-dnn*[2]*, ggml, Mlpack, OpenNN, ML.NET, Accord.NET, MLflow, etc)* [1]. It should also be noted that in addition to the control software (SW), the choice of hardware is influenced by the following factors: the type of neural network, the type of training, the training model[3], the size of the training sample (training data set), and the type of computational model itself. The share of application of neural networks for solving various problems is steadily increasing, and the requirements for the choice of hardware are changing accordingly. The

[1] NLTK (Natural Language Toolkit) is a library for natural language processing in Python.
[2] Tiny-dnn is a standalone C++ implementation of deep learning for use in low-computing environments.
[3] Machine learning model - algorithms that learn from data and are deployed to perform various tasks.

Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

weight of modern neural networks ranges from tens of megabytes to several terabytes, which accordingly sets the requirements for hardware. Tab. 1 shows the size of some «light» neural networks [2].

Table 1

Dimensionality of some neural networks

Model	Input size	Parameter size	Feature size	Model type	Dataset
rfcn-res50-pascal	600 x 850	122 MB	1 GB	resnet50	pascal VOC
rfcn-res101-pascal	600 x 850	194 MB	2 GB	resnet101	pascal VOC
ssd-pascal-vggvd-300	300 x 300	100 MB	116 MB	vvgd	pascal VOC
ssd-pascal-vggvd-512	512 x 512	104 MB	337 MB	vvgd	pascal VOC
ssd-pascal-mobilenet-ft	300 x 300	22 MB	37 MB	mobilenets	pascal VOC

When choosing hardware for neural networks, some of the following tasks can be identified

- the problem of assessing the required hardware resource for training a neural network with different types/sizes of training data;
- the problem of assessing the required hardware resource for training a neural network with a different number of layers in different training epochs;
- the problem of determining the dependence of the hardware resource on the type and numerical value of the hyperparameters[4]of neural networks;
- the problem of correlating the output values of different neural networks, when solving one problem, on a single hardware;
- the task of unified performance tests for hardware deployed to study the results of neural networks [4];
- the task of forming a correct request to the neural network (prompt[5]).

Different researchers and technology companies offer different solutions for choosing hardware for deploying and debugging neural networks[6]. Many different recommendations for choosing hardware are usually either highly specialized in nature, applicable to a specific neural network or, on the contrary, have a very broad scope. When deploying hardware for training neural networks, there is always the issue of performance testing. But unlike deterministic hardware systems, for which there are many unified tests and specialized distributions (software) [6,7,8,9,10] for performance evaluation, for hardware systems designed to work with ML there are no unified sets of tests [11,12,13,14], and research solutions are either advisory in nature or tied to a specific

[4] Hyperparameters are parameters of algorithms whose values are set before starting the learning process [3].
[5] Prompt - a request to the neural network to obtain the necessary data. The clearer and more correct the prompt is written, the more relevant the result will be. An incorrectly formulated prompt can overload both the neural network and the equipment.
[6] In particular, the Google Tensor Processing Unit [5].

Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

model/neural network. The most advanced progress in this area has been made by Harvard University research group, with a parameterized set of tests for various models and networks – ParaDnn [15], but this research also has several limitations. There is also no single reference book that includes the required hardware resource values when using a given single neural network (combination of networks). The absence of this reference book can be explained by the probabilistic model of neural networks functioning itself. The research [16] provides formulas for determining various types of neural network complexity *(structural, computational, computational complexity of the algorithm for synthesizing a neural network model)*, defines criteria for assessing the level of *logical transparency*[7]of a neural network, and compares neural network models and algorithms for their synthesis. Some of the formulas are used to determine various parameters of neural networks and to pre-calculate the required hardware resource within the given constraints. The analytical expressions are given below [16]:

- the computational complexity of the algorithm for synthesizing a neural network model is determined by the analytical expression 1,

$$N_{n.a.} = \eta_b S(N + N_M) + \eta_w N_w + \eta_v N_v \quad (1)$$

where, $N_{n.a.}$ - number of used memory cells, η_b - the number of memory cells for storing one element of training data, *N*- number of features characterizing instances, *S*- number of specimens in the sample, N_M - number of output features, η_w - number of memory cells for storing one network weight, N_w - number of weights and network thresholds, η_v - number of memory cells for storing one additional algorithm variable, N_v - number of algorithm variables.

- the computational complexity of the feedforward network in sequential implementation is determined by the analytical expression 2,

$$T_1 = \sum_{\mu=1}^{M} \sum_{i=1}^{N_\mu} T^{\mu,i} \quad (2)$$

where, M - number of network layers, N_μ - number of neurons in a μ layer.

- The computational complexity of the direct propagation network with hardware implementation of T_2 calculations is determined by analytical expression 3.
-

$$T_2 = \sum_{\mu=1}^{M} \underset{i}{Max} T^{(\mu,i)} \ , \quad i = 1,2,3,..., N_\mu \quad (3)$$

Also, important parameters are the logical transparency and redundancy of the network memory (K_1), which, for example, when researching malware, is one of the criteria for assessing the correctness of the training of the corresponding neural network. To determine the redundancy of the network, analytical expression 4 is used.

[7] A neural network that solves a problem in a way that is understandable to humans and for which it is easy to formulate a verbal description in the form of an explicit algorithm is called logically transparent.

Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

$$K_1 = \frac{N_w}{SN},\ N_w > 1,\ S > 0,\ N > 0 \quad (4)$$

If $K_1>1$, then the network memory is redundant (the dimension of the network memory is greater than the dimension of the training sample), if $K_1=1$, then the network can remember the entire training sample (the dimension of the network memory is equal to the dimension of the training sample), if $K_1<1$, then the network will not be able to remember the entire training sample exactly (the dimension of the network memory is less than the dimension of the training sample [16,17]. All the above analytical expressions are applicable to determine the preliminary value of the hardware resource with known parameters of the neural networks being trained. If the values of the parameters of the neural networks being researched are unknown, then the possibilities for accurately determining the required hardware resource are reduced. The selection of solutions/recommendations applicable to non-specialized hardware becomes an urgent task, since not all researchers have access to high-performance computing systems.

The novelty of the research lies in a more relevant attempt to catalog recommendations for the selection of hardware, with a priori known types of complexity of the neural networks being research.

Conflict Setting

It is necessary to formulate recommendations for choosing hardware resources when deploying neural networks.

Research results

Developing a set of recommendations for selecting hardware, with a brief description.

- **Central Processing Unit (CPU)**

When working with Models based on Parallel Computing (MPI[8]), the use of multi-core processors is mandatory. The size of the CPU cache does not play a big role - it is more important that the processor has a high-performance TLB (Translation lookaside buffer). But when running several neural networks simultaneously solving a single problem, the cache size is critical. Also, it is undesirable to use multiprocessor systems, since when several neural systems are launched simultaneously on a single dataset, even a minimal «memory leak» results in an uncontrolled increase in type 2 errors. Datasets for training neural networks are usually too large to fit in the cache, and for each new sample the data must be read from memory. Some ML libraries (like most applications) use only one thread (especially libraries implemented based on BLAS[9] operations). That is, the use of multi-core server processors is not always a priority. High CPU frequency is not always a priority parameter. Even with 100% CPU utilization, most of the utilization may come from fixing cache hit errors or TLB thrashing. At this stage of implementation, it is important to consider the type and complexity of the neural network (using analytical expressions 1,2,3). Accordingly, when setting up and configuring a system for ML, it is important to determine the tasks solved by neural networks in

[8] The Message Passing Interface (MPI) is a standardized and portable message passing standard designed to run on parallel computing architectures.

[9] Basic Linear Algebra Subprograms is a specification that prescribes a set of low-level routines for performing common linear algebra operations.

Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

the process of processing data during their circulation in the hardware. For large sample sizes and increased requirements for event response time, it is recommended to use Intel Itanium[10] microprocessors since, all other things being equal, these processors provide faster neural network restructuring due to higher parallelism of calculations and a large cache that provides data to each core at an average speed of several tens of Gb/s. Also, these processors always operate in x64-bit mode and have a 128-bit instruction format, unlike the x86_64 architectures, which switches between «Long»-x64 and «Legacy»-x86 modes. If possible, to solve highly specialized tasks *(Machine Vision, Natural Language Processing, etc)*, it is desirable to use specialized neural processors [18].

- **Graphics Processing Unit (GPU)**

Machine learning requires fast processing of large amounts of data. In addition to the number of cores, an important GPU parameter is the amount of memory. The GPU memory size determines how large a data pool (batch file) the GPU can process without fragmentation. This is especially true when working with piecewise contextual hashing data with a variable step, the size of which varies from a few bytes to several tens and hundreds of megabytes. The GPU must have such amount of memory that the condition - $K1 \geq 1$ is met. It is necessary that the entire data set and intermediate results can be accommodated in memory. (At this stage of implementation, the use of analytical expressions 3 and 4 for preliminary calculation is recommended). When reverse engineering models that have reached *«overfitting»*, an additional (25÷30) % memory resource is required to store reference values of weight coefficients. The use of tensor cores[11] significantly accelerates the model training process. The number of tensor cores is selected depending on specific tasks and the size of the training sample. Training models with millions of parameters require synchronous access to data. High throughput helps reduce latency when accessing model data. The higher the throughput, the faster the GPU can access, which is critical for many ML tasks, especially in systems that are elements of the Infrastructure Security System. In this case, the reaction time is one of the key parameters that can detect an external intrusion. The computing power of the GPU also affects the size of the batch file. The larger the batch file, the fewer similar operations the model will perform, which reduces the overall reaction time to the event.

- **Long-Term and Short-Term Memory**

Storing processed data on a hard drive is not always justified, since it is necessary to iteratively repeat read/write operations, the speed of which varies and is in any case lower than the speed of the CPU/GPU. It is advisable to have a created RAM disk[12] or use a Hard Disk Drive (HDD) with high rpm[13] (10000/14400). The use of Solid-State Drive (SSD) media is not recommended, since despite the high speed compared to HDD, there is a rapid wear of memory elements, which reduces both the reliability of the entire system and increases the number of errors in trained neural networks. Also, in the event of a physical failure of the SSD, it is quite difficult to recover data from it. To increase the

[10] Intel Itanium processors have been discontinued since 2019, but it is still possible to build a computing system on them.
[11] Tensor cores are specialized components of the graphics processor for performing operations on data arrays.
[12] The use of RAM (Random Access Memory, RAM) disk places special demands on ensuring uninterruptible power supply.

Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

overall read speed from the disk, you can combine the drives into a RAID 0 or RAID 10 (Redundant Array of Independent Disks, RAID) array, since in most cases RAID 10 offers better throughput and lower latency than all other RAID levels except RAID 0 (best throughput) [19]. It is recommended to use 4-channel RAM to increase the transfer rate between the memory controller and the CPU. The frequency and performance of the north bridge should be no lower than the same parameters of RAM, CPU, GPU and higher than the transfer rate of the system bus (Front Side Bus). To determine redundancy and prevent overloading of GPU or RAM memory, it is necessary to preliminarily conduct an assessment using expression (4) when deploying the system.

Properly selected hardware will improve quality, speed up training times, and reduce the response time of machine learning models.

Conclusion

The article considers a set of recommendations for selecting and configuring hardware when training neural networks. Some disadvantages and limitations inherent in different hardware components are formulated. A set of recommendations for selecting a central and graphic processor, long-term and short-term data devices is given. Some ML tasks are defined when configuring neural networks. Analytical expressions are given for determining the computational complexity of the neural network model synthesis algorithm, the computational complexity of the feedforward network with sequential implementation, the computational complexity of the feedforward network with hardware implementation of computations, and an expression for determining the network redundancy. Based on the recommendations described in the article, it is possible to more selectively select hardware intended for training neural networks and to perform a more granular configuration.

References

1. Douglas C. «A PyTorch Benchmark for High-Contrast Imaging Post Processing». https://doi.org/10.48550/arXiv.2409.16466
2. ITMO University Encyclopedia. «Machine Learning on Mobile Phones. The Weight of Modern Neural Networks». https://u.to/pe73IA , the resource is available on 28.10.2024.
3. Goodfellow I., Yoshua B., Courville A. Deep Learning, MIT, 2017, pp. 375-381.
4. Mitchell T. Machine Learning // McGraw-Hill Science/Engineering/Math, 1997, pp. 421.
5. Official page of Google cloud service. «Cloud Tensor Processing Units (TPUs)». https://cloud.google.com/tpu , the resource is available on 28.10.2024.
6. Akinshin A. Professional Benchmark. The Art of Measuring Performance // Apress, Piter 2022, St. Petersburg-Moscow-Minsk, p. 576.
7. Grigorieva M. et al. «High Energy Physics Data Popularity: ATLAS Datasets Popularity Case Study» // *Proceedings of the 2020 Ivannikov Memorial Workshop (IVMEM) IVMEM 2020, IEEE Computer Society*, p. 28-34.
8. PassMark software official page. https://www.passmark.com/ , the resource is available on 28.10.2024.
9. Official page of the UBCD. https://www.ultimatebootcd.com/ , the resource is available on 28.10.2024.
10. Hiren's BootCD PE official page. https://www.hirensbootcd.org/ , the resource is available on 28.10.2024.

[13] Revolutions Per Minute (RPM) - is a unit of rotational speed for rotating machines. One revolution per minute is equivalent to 1/60 Hz.

Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

11. Radulov N., Zhang Y., Bujanca M., Ye R., Lujan M. «A Framework for Reproducible Benchmarking and Performance Diagnosis of SLAM Systems». https://arxiv.org/abs/2410.04242
12. Jain K., Synnaeve G., Rozière B. «TestGenEval: A Real World Unit Test Generation and Test Completion Benchmark». https://arxiv.org/abs/2410.00752
13. Aleithan R. et al. «SWE-Bench+: Enhanced Coding Benchmark for LLM». https://doi.org/10.48550/arXiv.2410.06992
14. Dai X. et al. «How Do Large Language Models Understand Graph Patterns? A Benchmark for Graph Pattern Comprehension». https://doi.org/10.48550/arXiv.2410.05298
15. Wang Y., Wei G., Brooks D. «Benchmarking TPU, GPU, and CPU Platforms for Deep Learning». https://doi.org/10.48550/arXiv.1907.10701
16. Subbotin V. «Methodology and criteria for comparing models and algorithms for the synthesis of artificial neural networks», *Radioelectronics, informatics, management. 2003. №2 (10).* https://cyberleninka.ru/article/n/metodika-i-kriterii-sravneniya-modeley-i-algoritmov-sinteza-iskusstvennyh-neyronnyh-setey , the resource is available on 28.10.2024.
17. Qiu Y. MIBench: «A Comprehensive Benchmark for Model Inversion Attack and Defense». https://doi.org/10.48550/arXiv.2410.05159
18. Thibault S. «A Flexible Thread Scheduler for Hierarchical Multiprocessor Machines». https://doi.org/10.48550/arXiv.cs/0506097
19. Official website of Alterbit. RAID 10 (RAID 1+0). https://www.alterbit.ru/glossary34.html , the resource is available on 28.10.2024.

References

1. Douglas C. «A PyTorch Benchmark for High-Contrast Imaging Post Processing». https://doi.org/10.48550/arXiv.2409.16466
2. Энциклопедия университета ИТМО. «Машинное обучение на мобильных телефонах. Вес современных нейронных сетей». https://u.to/pe73IA , ресурс доступен на дату 28.10.2024.
3. Goodfellow I., Yoshua B., Courville A. Deep Learning, MIT, 2017, pp. 375-381.
4. Mitchell T. Machine Learning // McGraw-Hill Science/Engineering/Math, 1997, pp. 421.
5. Официальная страница облачного сервиса google. «Cloud Tensor Processing Units (TPUs)». https://cloud.google.com/tpu , ресурс доступен на дату 28.10.2024.
6. Акиньшин А. Профессиональный Бенчмарк. Искусство измерения производительности // Apress, Питер 2022, Санкт-Петербург-Москва-Минск, стр. 576.
7. Grigorieva M. et al. «High Energy Physics Data Popularity: ATLAS Datasets Popularity Case Study» // *Proceedings of the 2020 Ivannikov Memorial Workshop (IVMEM) IVMEM 2020, IEEE Computer Society*, p. 28-34.
8. Официальная страница ПО PassMark. https://www.passmark.com/ , ресурс доступен на дату 28.10.2024.
9. Официальная страница дистрибутива UBCD. https://www.ultimatebootcd.com/ , ресурс доступен на дату 28.10.2024.
10. Официальная страница дистрибутива Hiren's BootCD PE. https://www.hirensbootcd.org/ , ресурс доступен на дату 28.10.2024.
11. Radulov N., Zhang Y., Bujanca M., Ye R., Luján M. «A Framework for Reproducible Benchmarking and Performance Diagnosis of SLAM Systems». https://arxiv.org/abs/2410.04242
12. Jain K., Synnaeve G., Rozière B. «TestGenEval: A Real World Unit Test Generation and Test Completion Benchmark». https://arxiv.org/abs/2410.00752

Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

13. Aleithan R. et al. «SWE-Bench+: Enhanced Coding Benchmark for LLM». https://doi.org/10.48550/arXiv.2410.06992
14. Dai X. et al. «How Do Large Language Models Understand Graph Patterns? A Benchmark for Graph Pattern Comprehension». https://doi.org/10.48550/arXiv.2410.05298
15. Wang Y., Wei G., Brooks D. «Benchmarking TPU, GPU, and CPU Platforms for Deep Learning». https://doi.org/10.48550/arXiv.1907.10701
16. Субботин В. «Методика и критерии сравнения моделей и алгоритмов синтеза искусственных нейронных сетей», *Радіоелектроніка, інформатика, управління. 2003. №2 (10).* https://cyberleninka.ru/article/n/metodika-i-kriterii-sravneniya-modeley-i-algoritmov-sinteza-iskusstvennyh-neyronnyh-setey , ресурс доступен на дату 28.10.2024.
17. Qiu Y. MIBench: «A Comprehensive Benchmark for Model Inversion Attack and Defense». https://doi.org/10.48550/arXiv.2410.05159
18. Thibault S. «A Flexible Thread Scheduler for Hierarchical Multiprocessor Machines». https://doi.org/10.48550/arXiv.cs/0506097
19. Официальный сайти Alterbit. RAID 10 (RAID 1+0). https://www.alterbit.ru/glossary34.html , ресурс доступен на дату 28.10.2024.

ՆԵՅՐՈՆԱՅԻՆ ՑԱՆՑԵՐԻ ՈՒՍՈՒՑՄԱՆ ԺԱՄԱՆԱԿ ԱՊԱՐԱՏԱՅԻՆ ԱՊԱՀՈՎՄԱՆ ԸՆՏՐՈՒԹՅԱՆ ՎԵՐԱԲԵՐՅԱԼ ԱՌԱՋԱՐԿՈՒԹՅՈՒՆՆԵՐ

Թ.Վ. Ջամղարյան
Հայաստանի ազգային պոլիտեխնիկական համալսարան

Ներկայացված են նեյրոնային ցանցերը ուսուցանելու ժամանակ ապարատային ռեսուրսի ընտրության վերաբերյալ առաջարկություններ: Ձևակերպված են որոշ խնդիրներ, որոնք առաջանում են ապարատային ապահովման ընտրության և կազմաձևման ժամանակ, որի վրա ուսուցանվում են նեյրոնային ցանցերը: Տրվել է տարբեր նեյրոնային ցանցերի գրադարանների/ֆրեյմվորկերի կիրառման գնահատական: Ուսումնասիրվել են կենտրոնական ու գրաֆիկական պրոցեսորների ընտրության տարբերակները: Մշակվել են ապարատային բաղադրիչներ ընտրելու առաջարկություններ:

Բանալի բառեր. հիպերպարամետր, մեքենայական ուսուցման մոդել, ուսումնական հավաքածու, տրամաբանական թափանցիկություն, TensorFlow, Translation lookaside buffer, ParaDnn

Bulletin Of High Technology N 3 (31) 2024.-pp. 84-92.

T.V. Jamgharyan

RECOMMENDATIONS FOR THE DEPLOYMENT OF EQUIPMENT FOR TRAINING NEURAL NETWORKS

РЕКОМЕНДАЦИИ ПО РАЗВЕРТЫВАНИЮ АППАРАТНОГО ОБЕСПЕЧЕНИЯ ПРИ ОБУЧЕНИИ НЕЙРОННЫХ СЕТЕЙ

Т.В. Джамгарян
Национальный политехнический университет Армении

Представлены рекомендации по выбору аппаратного ресурса при обучении нейронных сетей. Рассмотрены аналитические выражения, для расчета вычислительной сложности при развертывании нейронных сетей прямого распространения различной размерности. Сформулированы некоторые задачи, возникающие при выборе и конфигурировании аппаратного обеспечения, на котором происходит обучение нейронных сетей. Дана оценка применения библиотек/фреймворков различных нейронных сетей. Выработан набор рекомендаций по взаимной компоновке модулей аппаратного обеспечения.

Ключевые слова: гиперпараметр, модель машинного обучения, обучающая выборка, логическая прозрачность, TensorFlow, Translation lookaside buffer, ParaDnn.

Submitted on 26.09.2024
Sent for review on 27.09.2024
Guaranteed for printing on 29.10.2024