UDC – 004.725:004.852

RESEARCH OF MODEL FOR INCREASE LEARNING SPEED OF A NEURAL NETWORK IN THE WINDOWS OPERATING SYSTEM

Timur V. Jamgharyan
National Polytechnic University of Armenia
105 Teryan St, 0009, Yerevan
e-mail: t.jamgharyan@yandex.ru
ORCID ID: 0000-0002-9661-1468
Republic of Armenia

https://doi.org/10.56243/18294898-2025.4-116

Abstract

The paper presents the effect of the processor time quantum size on the training speed of neural network. The Windows Server 2019 operating system was chosen as the operating environment, where the training speed of the neural network was analyzed with various system settings. A convolutional neural network was used as the object of research.To speed up the training of the neural network, a method based on the use of automatic command merging and a decrease in the processor time quantum size was used. The obtained results allow optimizing the training processes of neural networks in multitasking environments, ensuring an increase in the overall performance of the system. An analysis of the relationship between the parameters of planning computing resources and the training speed was carried out, and recommendations for optimizing systems taking into account the specifics of neural networks were proposed.

Keywords: batching, clockers, hyperparameter, convolutional neural network, failover cluster, operating environment, processor time quantum.

Introduction

The performance of neural network training heavily depends on the hardware. However, in addition to the architectural solutions of the equipment - the number of CPU (Central Processing Unit) and GPU (Graphics Processing Unit), the type and volume of long-term HDD/SSD (Hard Disk Drive, HDD/Solid-State Drive, SSD) and short-term (Random Access Memory, RAM) data storage devices, the transfer rate of the system bus, a parameter that often remains outside the attention of researchers of systems using machine learning (ML) - the quantum of processor time[1] - has a significant impact on the speed and stability of training. The processor time quantum is especially important in situations where the processor resources are divided between several tasks, such as when training a neural network under conditions of limited access to dedicated computing nodes. In such a case, the length of the processor time quantum determines how often the training task will have access to the processor and how smoothly sequential computations will be processed. If the time quantum is too small, the processor will frequently switch between tasks, creating

[1] Processor time quantum - this parameter defines the minimum time interval during which the processor allocates resources to perform a specific task.

additional load on the system due to context switches. This may lead to decreased performance and increased training time of the neural network. On the other hand, too large a time quantum may cause delays in processing I/O tasks. Thus, proper configuration of the processor time quantum becomes critical for optimal use of computing resources. For different operating environments, the value of the processor time quantum is different.

The values of the processor time quantum for the Windows operating systems (OS) that served as the platform for this research presented in Tab. 1 [1-3].

Table 1

The value of the quantum of processor time for Windows operating systems

Operating system	Quantum of time processor (ms)	Note
Windows 7	15–30	Depends on power mode. The scheduler is focused on the balance of interactivity and performance.
Windows 10	15–30	Uses a scheduler with dynamic quantum management. The interval changes depending on the system activity.
Windows 11	15–30	Similar to Windows 10, optimized for new processors and hybrid architectures.
Windows Server 2012, 2016	120	Optimized for server tasks with minimal context switching frequency.
Windows Server 2019	120	Expanded support for containers and cloud solutions, optimization for long-term calculations.
Windows Server 2022	120	Security and performance improvements for large server workloads.

When training neural networks on computers united in clusters, it is important to correctly configure the processor time quantum depending on both the type of neural network being trained and its parameters/hyperparameters[2].

A pressing task is to determine the optimal time quantum value to increase the learning/response speed in parallel computations, such as distributed training of a neural network (networks) on several nodes. Researchers in ML systems offer various solutions to this problem [6-9] each of which has certain limitations. However, the method of accelerating the training of neural networks by changing the automatic combination of commands (batching[3]) with a parallel change in the quantum of the processor has not been considered.

The novelty of the research lies in the application of methods for changing the batching size and the parameters of the quantum of the processor to increase the speed of training of a convolutional neural network.

Conflict Setting

It is necessary to obtain the optimal value of the processor time quantum for a given operating system when training a neural network.

[2] Hyperparameters of a neural network are parameters of a ML model that are set before training begins. Hyperparameters, unlike parameters, are unchanged during the training process [4, 5].

[3] Batching (ML) is changing multiple weights in different layers of neural networks with one command.

Discussion

By changing the values of the processor time quantum for a given OS, as well as modifying the batching value for the convolutional neural network code, obtain the most effective value for the learning speed.

Boundary condition

- at a given point in time, only one neural network is trained,
- training was carried out based on only the CPU, on computers united in a cluster.

Experimental procedures

A computing cluster[4] (Fig. 1) consisting of 12+1 computers was deployed under Windows Server 2019 Standard (build 17763). The Failover cluster [10] and Hyper-V [11] roles are activated and configured in the OS. The computing cluster is configured using the SAN (Storage Area Network, SAN) architecture.

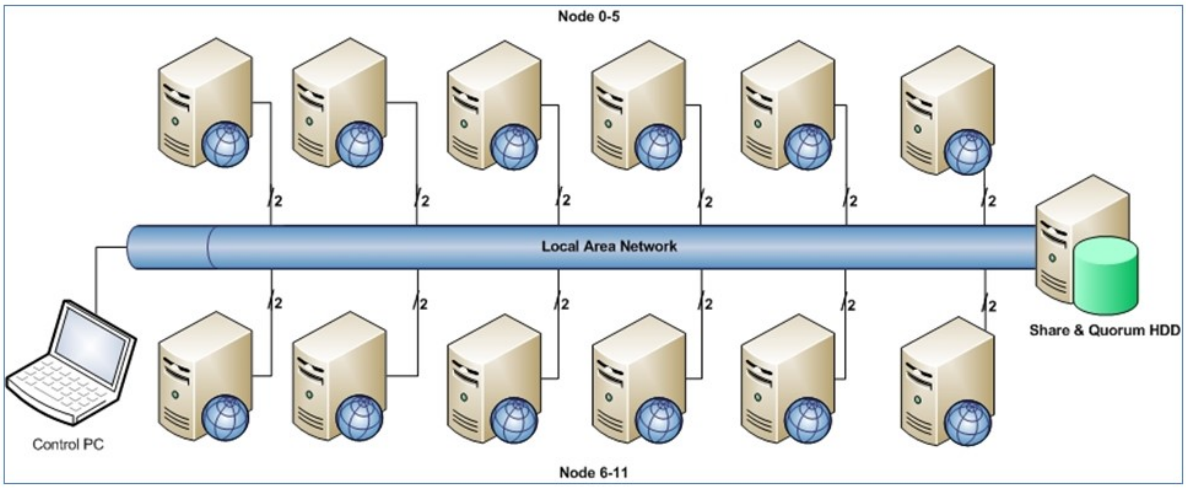

Fig. 1 Diagram of the computing cluster

The computers are connected two local networks (control and data transmission) using two network interfaces. A common storage is implemented on a separate computer and a quorum disk is activated. The OS has Tensor Flow software [12] installed, in which a convolutional neural network is deployed. The processor time quantum was changed according to the description presented in [13]. The control of the change in the quantum of processor time for the Windows OS was carried out by the *Clockres* software [14, 15].

Research was carried out with different values of the hyperparameters of the convolutional neural network. Fig. 2 show the modified values of the processor time quantum for the Windows Server 2019 OS used in the research.

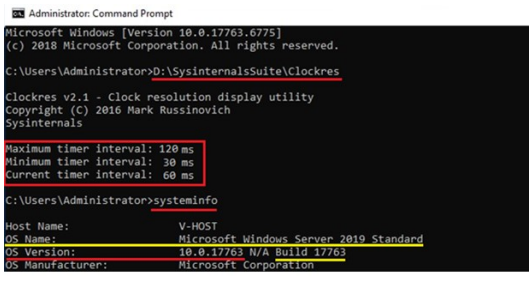

Fig. 2 Processor time quantum value for Windows Server 2019 OS

The processor time quantum was changed in the range of 100÷30 ms (the value of 30 ms corresponded to the upper value of the client version of Windows OS). In all cases, testing was

[4] Parameters of a single computer: CPU-i9, RAM 16Gb.

performed with the security software disabled and the OS updated to the current state [16]. The batching of processes was changed separately based on the developed script.

Training was carried out in three stages:

- training with the standard value of the processor time quantum with the batching change script disabled,
- training with the changed value of the processor time quantum, with the batching change script disabled,
- training with the changed value of the processor time quantum with the batching change script enabled.

The learning rate was compared with the learning rate at the standard, upper value of the processor time quantum (120 ms) for the given OS.

The reliability of the output values of the trained neural network was checked by comparing it with the a priori known output values of the neural network trained in the research [17] (reference network).

The comparison was performed by WinMerge software [18].

Research Results

The research results change in learning rate at different values of the processor time quantum are presented in Tab. 2. The dependence of the learning rate on the value of the processor time quantum is shown in Fig. 3.

Table 2

Results of the research on changes in learning speed

OS	Value of quantum of processor time (ms)	Convolutional neural network learning rate increment*(%) / confidence (%)			Convolutional neural network learning rate increment*(%) / confidence (%)		
		Number of layers			Number of layers		
		3	5	7	3	5	7
Windows Server 2019	100	0.6/97	0.5/98	0.2/95	0.9/96	0.7/92	0.4/97
	90	0.8/95	0.4/90	0.4/88	0.8/98	1.0/92	0.8/93
	80	1.3/96	0.9/91	0.6/81	1.7/95	1.6/88	0.8/90
	70	1.8/85	1.3/90	0.8/88	2.5/92	1.9/90	1.2/87
	60	2.4/97	1.6/85	0.8/83	3.7/89	2.6/91	1.8/89
	50	3.7/96	2.3/88	0.9/85	4.3/91	2.9/94	1.9/92
	40	3.2/94	2.1/90	1.1/82	4.0/88	2.3/89	1.5/89
	30	2.6/92	1.9/89	1.0/82	3.7/91	2.2/92	1.3/90

* learning speed with batching disable,

** learning speed with batching enable.

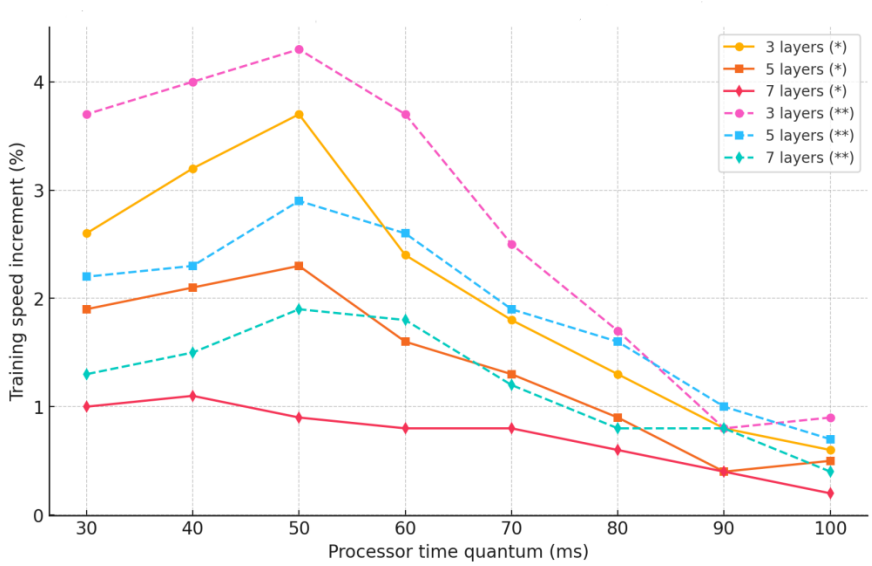

Fig. 3 Dependence of the learning rate on the quantum of processor time

Based on the experiment, the following conclusions can be drawn:

- the optimal value of the processor time quantum for Windows Server 2019 OS when training a convolutional neural network is in the range of (40÷60) ms,
- the increase in the learning rate at the optimal value is (3.7÷4.3)%,
- reliability, with an increase in the learning rate, decreases on average by (3÷13)%.

Conclusion

The paper considers a model for accelerating the training of a convolutional neural network based on a hardware cluster with a variable value of the quantum of processor time and software optimization. The change in the quantum of processor time was controlled by the *Clockres* utility from the *Sysinternals* software package.

It was determined that in all cases, when the quantum of processor time is reduced, the training speed increases, with a parallel decrease in the reliability of output values. Using a large quantum of time allows ensuring the stability of long-term computing processes.

For deep networks (7+ layers), it is recommended to use server OS. For small and medium networks with less intensive calculations, client OS or server OS with a minimum quantum of processor time may be optimal.

References

1. Russinovich M., Yosifofovich P., et al., Windows Internals, Part 1: System architecture, processes, threads, memory management, and more, 7th Edition., The Microsoft Press Store By Pearson, 2018, pp. 945
2. Minasi M., et al, Mastering Windows Server 2012R2, Part:1. SYBEX, 2015, pp. 950
3. Microsoft Learn. Measure application performance by analyzing CPU utilization (C#, Visuall Basic, C++, F#), description page https://learn.microsoft.com/en-us/visualstudio/profiling/beginners-guide-to-performance-profiling?view=vs-2022 , the resource is available on 25.01.2025

4. Goodfellow Y., Bengo Y., Courville A., Deep Leaning, MIT Press, 2017, pp. 800.
5. Breuel T., «The Effects of Hyperparameters on SGD Training of Neural Networks». https://doi.org/10.48550/arXiv.1508.02788
6. Hazem T., Abdelhadi A., «Hardware Proximal Policy Optimization – A Universal Pipelined Architecture for Generalized Advantage Estimation». https://doi.org/10.48550/arXiv.2501.12703
7. Ryu M., Byeon G., Kim K., «A GPU-Accelerated Distributed Algorithm for Optimal Power Flow in Distribution Systems». https://doi.org/10.48550/arXiv.2501.08293
8. Wang Z., et al, «Dumpy OS: A Data-Adaptive Multi-ary Index for Scalable Data Series Similarity Search». https://doi.org/10.48550/arXiv.2501.08293
9. Cheng C., et al, «Dynamic Optimization of Storage Systems Using Reinforcement Learning Techniques». https://doi.org/10.48550/arXiv.2501.00068
10. Microsoft Learn. Failover Clustering in Windows Server and Azure Local description page https://learn.microsoft.com/en-us/windows-server/failover-clustering/failover-clustering-overview , the resource is available on 20.03.2025
11. Microsoft Learn. Software Defined Networking (SDN) description page https://learn.microsoft.com/en-us/windows-server/networking/sdn/ , the resource is available on 20.03.2025.
12. Tensor Flow software download official page. https://www.tensorflow.org/install?hl=ru , the resource is available on 20.03.2025
13. Microsoft Learn. Thread and task architecture guide, description page https://learn.microsoft.com/en-us/sql/relational-databases/thread-and-task-architecture-guide?view=sql-server-ver16 , the resource is available on 20.03.2025.
14. Russinovich M., Margosis A., Windows Sysintenals Administrator's Reference, Microsoft Press, 2012, pp. 478
15. Windows operating systems testing tools official download page, https://learn.microsoft.com/en-us/sysinternals/ , the resource is available on 20.03.2025
16. Official download page for various versions of the Windows operating system, https://www.microsoft.com/en-us/evalcenter , the resource is available on 20.03.2025.
17. Jamgharyan T., Iskandaryan V., Khemchyan A., (2024). Obfuscated Malware Detection Model. *Mathematical Problems of Computer Science*, *62*, 72–81. https://doi.org/10.51408/1963-0122
18. WinMerge software official download page, https://winmerge.org/downloads/?lang=ru , the resource is available on 20.03.2025

References

1. Russinovich M., Yosifofovich P., et al., Windows Internals, Part 1: System architecture, processes, threads, memory management, and more, 7th Edition., The Microsoft Press Store By Pearson, 2018, pp. 945
2. Minasi M., et al, Mastering Windows Server 2012R2, Part:1. SYBEX, 2015, pp. 950
3. Microsoft Learn. Measure application performance by analyzing CPU utilization (C#, Visuall Basic, C++, F#), description page https://learn.microsoft.com/en-us/visualstudio/profiling/beginners-guide-to-performance-profiling?view=vs-2022 , the resource is available on 25.01.2025
4. Goodfellow Y., Bengo Y., Courville A., Deep Leaning, MIT Press, 2017, pp. 800
5. Breuel T., «The Effects of Hyperparameters on SGD Training of Neural Networks». https://doi.org/10.48550/arXiv.1508.02788

6. Hazem T., Abdelhadi A., «Hardware Proximal Policy Optimization – A Universal Pipelined Architecture for Generalized Advantage Estimation». https://doi.org/10.48550/arXiv.2501.12703
7. Ryu M., Byeon G., Kim K., «A GPU-Accelerated Distributed Algorithm for Optimal Power Flow in Distribution Systems». https://doi.org/10.48550/arXiv.2501.08293
8. Wang Z., et al, «Dumpy OS: A Data-Adaptive Multi-ary Index for Scalable Data Series Similarity Search». https://doi.org/10.48550/arXiv.2501.08293
9. Cheng C., et al, «Dynamic Optimization of Storage Systems Using Reinforcement Learning Techniques». https://doi.org/10.48550/arXiv.2501.00068
10. Microsoft Learn. Failover Clustering in Windows Server and Azure Local description page https://learn.microsoft.com/en-us/windows-server/failover-clustering/failover-clustering-overview , the resource is available on 20.03.2025
11. Microsoft Learn. Software Defined Networking (SDN) description page https://learn.microsoft.com/en-us/windows-server/networking/sdn/ , the resource is available on 20.03.2025
12. Официальная страница загрузки ПО Tensor Flow. https://www.tensorflow.org/install?hl=ru , the resource is available on 20.03.2025
13. Microsoft Learn. Thread and task architecture guide, description page https://learn.microsoft.com/en-us/sql/relational-databases/thread-and-task-architecture-guide?view=sql-server-ver16 , the resource is available on 20.03.2025
14. Russinovich M., Margosis A., Windows Sysintenals Administrator's Reference, Microsoft Press, 2012, pp. 478
15. Официальная страница загрузки инструментов тестирования операционной системы Windows, https://learn.microsoft.com/en-us/sysinternals/ , the resource is available on 20.23.2025
16. Официальная страница загрузки различных версий операционной системы Windows, https://www.microsoft.com/en-us/evalcenter , the resource is available on 20.03.2025
17. Jamgharyan T., Iskandaryan V., Khemchyan A., (2024). Obfuscated Malware Detection Model. *Mathematical Problems of Computer Science*, *62*, 72–81 https://doi.org/10.51408/1963-0122
18. Официальная страница загрузки ПО WinMerge, https://winmerge.org/downloads/?lang=ru , the resource is available on 20.03.2025

ՆԵՅՐՈՆԱՅԻՆ ՑԱՆՑԻ ՈՒՍՈՒՑՄԱՆ ԱՐԱԳՈՒԹՅԱՆ ԲԱՐՁՐԱՑՄԱՆ ՄՈԴԵԼԻ ՀԵՏԱԶՈՏՈՒԹՅՈՒՆ WINDOWS ՕՊԵՐԱՑԻՈՆ ՀԱՄԱԿԱՐԳՈՒՄ

Թ.Վ. Ջամղարյան
Հայաստանի ազգային պոլիտեխնիկական համալսարան

Հոդվածում ներկայացված է պրոցեսորի ժամանակի քվանտի չափի փոփոխման ազդեցությունը նեյրոնային ցանցի ուսուցման արագության վրա: Փորձերն անցկացվել են Windows Server 2019 Standard օպերացիոն համակարգի միջավայրում: Հետազոտության արդյունքները ցույց են տալիս, որ բատչինգի մեթոդի կիրառումը և պրոցեսորի ժամանակի քվանտի չափը նվազեցնելն արագացնում է փաթույթային նեյրոնային ցանցի ուսուցումը հաշվողական ռեսուրսների ավելի ռացիոնալ բաշխման շնորհիվ: Ստացված արդյունքները թույլ են տալիս լավարկել նեյրոնային ցանցերի ուսուցման գործընթացները բազմաֆունկցիոնալ միջավայրերում:

T.V. Jamgharyan

RESEARCH OF MODEL FOR INCREASE LEARNING SPEED
OF A NEURAL NETWORK IN THE WINDOWS OPERATING SYSTEM

Բանալի բառեր. *հիպերպարամետր, օպերացիոն միջավայր, փաթույթային նեյրոնային ցանց, պրոցեսորի ժամանակի քվանտ, batching, clockres,* failover cluster*:*

ИССЛЕДОВАНИЕ МОДЕЛИ ПОВЫШЕНИЯ СКОРОСТИ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ В ОПЕРАЦИОННОЙ СИСТЕМЕ WINDOWS

Т.В. Джамгарян
Национальный политехнический университет Армении

В статье рассматривается влияние размера кванта времени процессора на скорость обучения нейронных сетей. В качестве операционной среды выбрана операционная система Windows Server 2019, где анализировалась скорость обучения нейронной сети при различных настройках системы. Для ускорения обучения нейронной сети использовался метод на основе применения автоматического совмещения команд и уменьшение размера кванта времени процессора. Проведен анализ взаимосвязи между параметрами планирования вычислительных ресурсов и скоростью обучения, а также предложены рекомендации по оптимизации систем с учётом специфики работы нейронных сетей.

***Ключевые слова*:** гиперпараметр, операционная среда, квант времени процессора, сверточная нейронная сеть, batching, clockres, failover cluster.

Submitted on 11.02.2025
Sent for review on 26.02.2025
Guaranteed for printing on 29.12.2025