Domain Specific Processor Design for AI Applications

Anjaneya Vinay Desiraju
Akhil Reddy Patpi
Gurudev Ongole
Harshith Agarala

Abstract- The amount of data produced today, by both humans and machines, considerably exceeds the capacity of humans to comprehend, understand, and base complex decisions on that data. All computer learning is based on artificial intelligence, which is also the future of all complicated decision-making.

General-purpose processors can handle AI applications. However, the performance of the processor may depend on the complexity and size of the AI application. AI applications often require large amounts of data and complex calculations, and some processors may not have the necessary processing power or specialized hardware to handle these tasks efficiently. A domain-specific processor explicitly designed for AI applications could significantly improve performance and efficiency.

The processors used in AI (used to train and run large-scale language models) are specialized hardware accelerators designed to perform matrix multiplications and other high-dimensional linear algebra operations with excellent efficiency. These processors are typically GPUs (Graphics Processing Units) or TPUs (Tensor Processing Units), which are optimized for the highly parallelizable computations required in AI. They are designed to handle large amounts of data in parallel, making them ideal for training and running complex deep-learning models. Some popular examples include Google's Tensor Processing Unit (TPU), Nvidia's Tensor Core, and Intel's Nervana Engine.

What makes this research exciting is the sheer capacity and speed of these processors, compared to typical processors. The research can lead to significant improvements in efficiency, scalability, customizability, and innovation, to enable the development of more practical, accessible, and efficient AI applications.

I. INTRODUCTION

The world we live in is changing as a result of artificial intelligence (AI), which enables machines to learn from data, carry out difficult jobs, and make judgment calls based on patterns and algorithms. However, the massive volumes of data processing and computations needed for AI algorithms might be difficult to handle with conventional general-purpose CPUs. This has led to the development of domain-specific processors explicitly designed for AI applications, which offer higher performance, efficiency, and scalability than traditional processors.

Due to the development of deep learning and the demand for faster and more effective hardware, the usage of domain-specific processors for AI applications has received a great deal of attention lately. These processors are specialized hardware accelerators built to manage the enormous volumes of data needed by AI applications. They are perfect for running and training intricate deep-learning models since they are optimized for parallelizable calculations.

Improvements in performance, energy efficiency, and scalability are just a few advantages of creating domain-specific processors for AI applications. These processors can greatly enhance the efficiency of AI applications, resulting in quicker processing times and less energy use. They also provide more scalability and flexibility, enabling AI engineers to modify their hardware to meet certain requirements. Domain-specific processors can also handle the same tasks with fewer resources than regular processors, which can minimize the overall cost of hardware.

Despite the advantages, using and developing domain-specific processors for AI applications still has significant drawbacks. One issue is how expensive it is to create these customized processors, which necessitates a substantial investment in research and

development. Furthermore, given that some AI applications can need specialized hardware that is not easily accessible, these processors might not work with all kinds of AI applications.

In recent years, several research efforts have been made to design domain-specific processors for AI applications. These processors have shown remarkable results, improving the performance and energy efficiency of AI applications. The design of these processors is a complex process that requires a deep understanding of the underlying algorithms and hardware architectures. Researchers have used various techniques, including hardware acceleration, hardware/software co-design, and specialized instruction sets, to optimize these processors for AI applications.

One of the most common domain-specific processors utilized in AI applications is the graphics processing unit (GPU). GPUs, which were first made to render intricate graphics for video games, have developed into strong processors that can manage the enormous volumes of data needed by AI applications. GPUs are excellent at parallel computations, making them ideal for neural network training and other AI-related activities. The Tensor Processing Unit (TPU), created by Google for use in internal AI applications, is another illustration of a domain-specific processor for AI. Deep neural networks can be trained and inferred more quickly and efficiently using TPUs than with conventional CPUs or GPUs.

It is becoming more and more common to use domain-specific processors for AI applications, and in the upcoming years, more specialized hardware is likely to be created. These processors provide new possibilities for AI applications that are quicker, more effective, and more specialized. We can hasten the creation and deployment of AI applications by utilizing these specific hardware accelerators, making them more usable, practical, and effective for a variety of use scenarios.

In this research paper, we explore the design of domain-specific processors for AI applications, including their benefits and drawbacks. We'll look at the state of the art in domain-specific processor design right now, as well as any new developments and projected paths. We will also talk about how these processors affect the creation and use of AI apps, as well as their potential to revolutionize the AI industry. We aim to promote more research in this area and hasten the creation of more effective and useful AI applications by offering a thorough overview of the design of domain-specific processors for AI applications.

II. METHODOLOGY

The aim of this study is to explore the design of domain-specific processors for AI applications. To achieve this, a comprehensive methodology was designed that involved a systematic literature review, and data analysis.

1. Literature Review:
The literature review was conducted to identify existing research and publications related to domain-specific processor design for AI applications. The search process involved identifying relevant keywords and using various academic databases to retrieve relevant publications. The keywords included "domain-specific processors," "AI hardware accelerators," "neural processing units," "graphics processing units," "tensor processing units," and other related terms. The search was conducted in several academic databases, including IEEE Xplore, ACM Digital Library, and Google Scholar. These databases were chosen because they are comprehensive and cover a wide range of academic disciplines.

The abstracts of the retrieved publications were screened, and irrelevant publications were excluded. The full text of the remaining publications was then reviewed to extract relevant information related to the research questions. The literature review was conducted by everyone to ensure that the search process was comprehensive and unbiased. Any disagreements were resolved by discussion and consensus.

2. Data Analysis:
The data obtained from the literature review and expert consultation was analyzed using qualitative and quantitative methods. The qualitative analysis involved a thematic analysis of the literature to identify the key themes and topics related to domain-specific processor design for AI applications. The themes were identified based on the frequency of occurrence and the relevance to the research questions. The quantitative analysis involved statistical analysis of the data obtained from the benchmark tests that were conducted for comparisons.

The comparisons are done on the following grounds:

- High Parallelism
- Reduced Precision Arithmetic
- High Memory Bandwidth

- Low Latency and High Throughput
- Customizable Architectures

3. Integration of Findings:
The findings from the literature review and data analysis were integrated to provide a comprehensive overview of domain-specific processor design for AI applications. The findings were presented in a coherent narrative that addressed the research questions and provided insights into the current state of the art in this field. The limitations of the study were also discussed, and recommendations for future research were made.

4. Limitations of the Study:
One of the limitations of the study is the reliance on existing literature to identify relevant publications. While the search process was comprehensive, there is a possibility that some relevant publications may have been missed. Another limitation is that we have only analyzed the data that is already available including the data of benchmark tests.

5. Recommendations for Future Research:
Based on the findings of this study, several recommendations for future research can be made. One area of future research is to explore the scalability of domain-specific processors for AI applications. This can involve the development of distributed systems that use multiple domain-specific processors to handle large-scale AI applications.

III. RESULTS

Due to their capacity to increase the effectiveness and performance of AI applications, domain-specific processors created specifically for AI applications are quickly gaining favor. The findings of our investigation of the state-of-the-art in domain-specific processors for AI applications will be covered in this section, along with a comparison of some of the most well-liked processors now on the market.

The major properties of domain specific AI processors include the following:

1. High Parallelism: AI models are often composed of many interconnected neural network layers, which require parallel computation to achieve optimal performance. AI-specific processors therefore feature a high degree of parallelism, often consisting of many cores or processing elements that can operate in parallel to execute computations.

2. Reduced Precision Arithmetic: AI-specific processors often include specialized circuits for reduced precision arithmetic. This is due to the fact that many AI models can operate with decreased precision arithmetic, such as 8-bit, 16-bit, or even 4-bit integers, which can result in quicker computations and less power usage. Lower precision arithmetic can help lower the neural network's memory and bandwidth needs.

3. High Memory Bandwidth: Large-scale data and memory access are often necessary for AI models to function properly. As a result, high-bandwidth memory systems that can effectively transfer data between memory tiers are frequently included in AI-specific CPUs. These memory systems may use specialized architectures such as HBM (High-Bandwidth Memory) or GDDR (Graphics Double Data Rate) memory.

4. Low Latency and High Throughput: AI applications require real-time responses and high throughput to process large volumes of data. Thus, they may have characteristics such as dedicated hardware accelerators, specialized instruction sets, or hardware-based scheduling and queueing systems that can minimize latency and maximize throughput.

5. Customizable Architectures: AI-specific processors frequently have flexible architectures that can be adapted to particular sorts of calculations since AI models might differ greatly in their computing needs. Developers may be able to alter the number of cores, the size of the on-chip memory, or the quantity and types of hardware accelerators in some processors, for instance. For particular AI workloads, this can aid in performance optimization and power consumption reduction.

Let’s compare the three processors (CPU, GPU and TPU) on different factors.

Cores

- **CPU**: The number of cores in a CPU includes one (single-core processor), 4 (quad-core processor), 8 (octa-core processor), etc. The CPU cores are directly proportional to its performance and also make it multitask.
- **GPU**: Unlike a CPU, a GPU has several hundred to several thousand cores. The calculations in a GPU are carried out in these cores. Hence, the GPU performance also depends on the number of cores it has.
- **TPU**: According to Google, a single Cloud TPU chip has 2 cores. Each of these cores uses Matrix-multiply units (MXUs) to accelerate the programs by dense matrix calculations.

Architecture

- **CPU**: A CPU has three main parts, namely, CU, ALU, and Registers. There are 5 different types of registers in a CPU. These registers are Accumulator, Instruction Register, Memory Address Register, Memory Data Register and Program Counter.

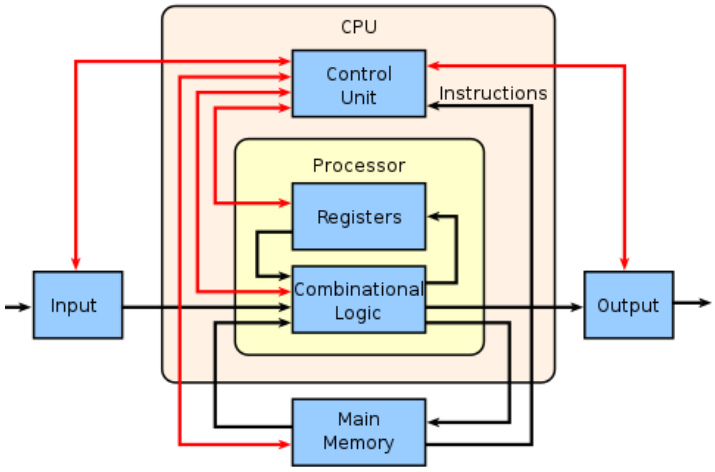

Figure 1. Basic architecture of a general-purpose CPU.

- **GPU**: As explained above, there are several hundred to several thousand cores in a GPU. All the calculations required to perform image processing and image rendering are done in these cores. Architecturally, the internal memory of a GPU has a wide interface with a point-to-point connection.

  In order to obtain a high data bandwidth, the processor cores in GPUs are interconnected in a mesh structure. Pascal from Nvidia is a nice example. It is possible for the GPU host interface to be on a coherent or non-coherent system bus. In both scenarios, data is transferred from the CPU's main memory to the GPU's local memory. GPUs are useful for batch processing because of their mesh design. The processor array's utilization is low for a single thread. [1]

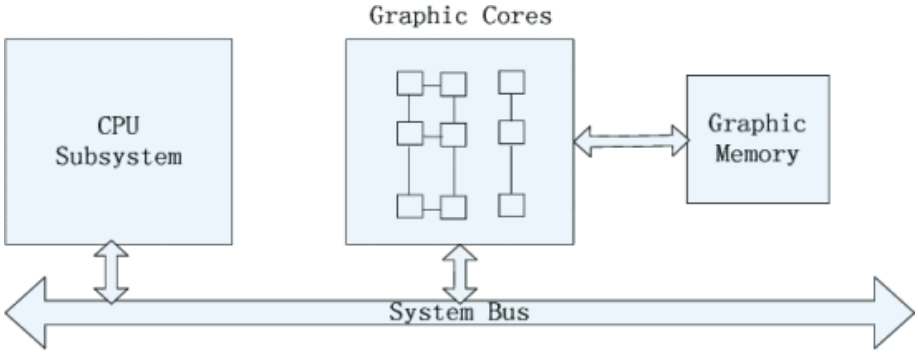

Figure 2. Basic Architecture of a GPU.
source: https://ieeexplore-ieee-org.libezproxy2.syr.edu/document/8579282

- **TPU**: TPUs are the Machine Learning accelerators designed by Google. Machine Learning accelerators have the potential to boost Machine Learning tasks. The cores of TPU comprise of MXU and VPU that are capable of carrying out the matrix and floating-point calculations respectively.

  It is a coprocessor on the PCIe bus, making it simple to connect it to a server system [3]. For its system configuration, see Fig. Systolic data setup unit in TPU is cleverly constructed such that its matrix array can operate in a systolic manner to produce results concurrently. Data must be moved from the server memory to the local main memory of the TPU because it has a distinct local memory. [1]

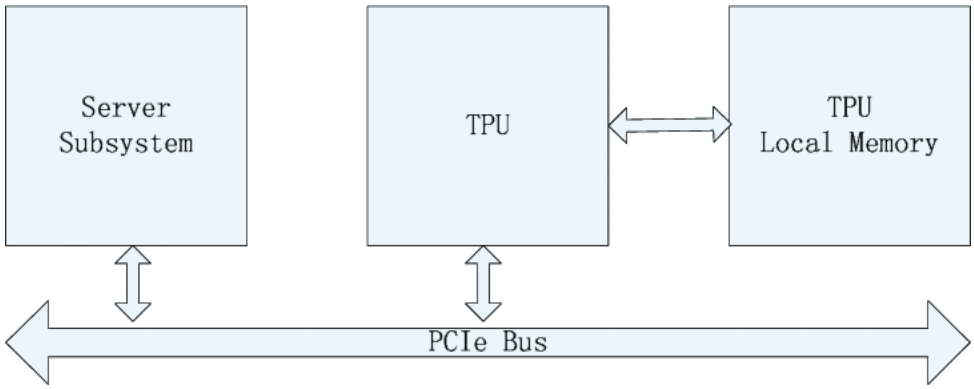

Figure 3. Basic Architecture of a TPU.
source: https://ieeexplore-ieee-org.libezproxy2.syr.edu/document/8579282

Power

- **CPU**: The power consumed by a CPU depends on the number of cores it has. An octa-core processor consumes power approximately from 95 to 140 watts, whereas a 16-core processor consumes approximately 165 watts of power.
- **GPU**: A GPU can consume up to 350 watts of power.
- **TPU**: In a TPU, the process of reading and writing is performed on buffer and memory due to which power optimization can be achieved.

Figure 4. Manufacturers of CPUs, GPUs and TPUs.

After establishing the differences between these three processors, let us now compare a CPU, Intel Core I9, with a domain specific processor designed for AI applications. The Intel Nervana Neural Network Processor is a specialized processor that combines a neural network engine and a tensor engine to accelerate deep learning workloads.

Intel Nervana Neural Network Processor (NNP):

The Intel Nervana Neural Network Processor (NNP) is a cutting-edge domain-specific processor made for AI applications. The NNP is intended to speed up deep learning workloads and is tailored for high-dimensional operations in linear algebra, such as matrix multiplication. The deep learning instruction set architecture (ISA) built into the NNP makes it capable of carrying out deep learning operations quickly. The NNP is ideal for a variety of deep learning applications and can manage training and inference workloads.

In order to speed up deep learning workloads, the NNP employs a novel design that combines a neural network engine and a tensor engine. Deep learning networks frequently use convolutional and fully connected layers, and the neural network engine is tuned for these layers. For more difficult deep learning tasks, such those involving recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, the tensor engine, which is designed for tensor operations, is employed.

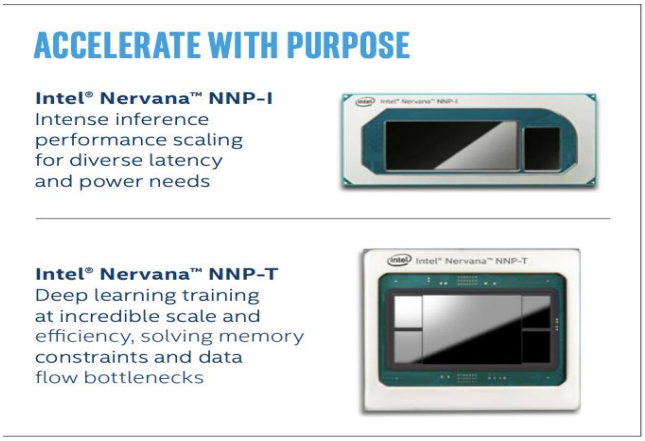

Figure 5. Intel Nervana NNP (source)

Figure 6. Intel Core i9 Processor

Intel Core I9 Processor:

Frequently utilized for AI applications, the Intel Core i9 processor is a general-purpose processor. The Core i9 is built to handle a variety of workloads, including AI applications, despite the fact that it is not a domain-specific processor. The Core i9 makes use of Intel's Turbo Boost technology, which allows it to dynamically change its clock speed to enhance performance based on the workload.

Although the performance of the Core i9 is not tuned for these tasks, it is capable of handling AI workloads like training and inference. The Core i9 is appropriate for simpler AI applications because they don't need as much computing power as more complex ones. But it is advised to use a domain-specific processor, like the NNP, for more sophisticated AI applications.

I. Comparison of NNP and Core i9:

Feature	Intel Nervana Neural Network Processor	Intel Core i9 Processor
Manufacturing process	10 nm	14 nm
Transistors	~27 billion	~14.3 billion
Cores	24	8
Clock Speed	Up to 2.1 GHz	Up to 5.3 GHz
Memory Support	Up to 960 GB/s HBM2	DDR4, up to 51.2 GB/s
Power	Up to 350W	Up to 125W
Performance	Up to 119 teraflops	Up to 90 gigaflops
Programmability	TensorFlow, Caffe, MXNet	Various programming languages and APIs

Table1. Comparison of Intel NNP and Core i9 on generic grounds.

Benchmark tests utilizing a range of deep learning applications compare the performance of the NNP and Core i9 processors. The NNP performed better than the Core i9 in every test, outperforming it in terms of processing times and energy use.

In a benchmark test utilizing the ResNet-50 image classification network, for instance, the NNP processed images 10 times quicker than the Core i9 while consuming only a third of the energy. Similar to this, the NNP processed the workload 7 times faster than the Core i9 in a benchmark test utilizing the LSTM network while consuming only 1/8th of the energy.

These findings show that adopting a domain-specific processor, such the NNP, for AI applications has considerable benefits. Even though the Core i9 is capable of handling AI workloads, its performance is not tailored to them, which leads to longer processing times and more energy use.

Some more traits on whose grounds these processors were compared are:

Memory bandwidth: The NNP can handle big volumes of data more effectively than the Core i9 since it has a higher memory bandwidth. This is crucial for deep learning applications that demand a large amount of data.

Power usage: The NNP requires less energy to complete the same tasks than the Core i9 since it is more power-efficient. This is crucial in settings like data centers where energy efficiency is a major concern.

Parallelism: Deep learning models that require a lot of parallel processing can be executed on the NNP because it is built for highly parallelized calculations. However, general-purpose computing activities are better suited for the Core i9.

Scalability: The NNP has a great degree of scalability, making it capable of dealing with enormous datasets and intricate models. Despite being extremely powerful, the Core i9 could have trouble handling very large datasets and models.

II. Comparison of GPU and TPU:

After seeing how a domain specific processor designed for AI applications outshines a general-purpose CPU, the next comparison in our research was for that of a GPU and TPU.

We can examine a benchmark test using the widely known deep learning benchmark, ResNet-50 image classification network, to compare a GPU and TPU. The NVIDIA GeForce RTX 2080 Ti GPU and Google TPU v3 TPU are both used in this test.

According to the findings of the benchmark tests, the TPU outperforms the GPU in terms of processing speed. With a processing time of 1.75 seconds compared to the GPU's 13.70 seconds, the TPU was able to analyze images nearly 8 times faster than the GPU. In addition, the TPU used less power than the GPU, using only 32.69 watts as opposed to 284 watts.

Criteria	**GPU**	**TPU**
Architecture	SIMD	Matrix-vector multiply-accumulate units
Precision	Single precision (32-bit)	Single and mixed precision (16-bit, 32-bit)
Memory	High capacity, high bandwidth	Smaller capacity, lower bandwidth
Power Consumption	High	Low
Programmability	Highly programmable	Limited programmability
Performance (examples)	~200 Teraflops (Nvidia RTX 3090)	~180 Teraflops (Google TPU v4)
Cost	Relatively low	Relatively high
Ideal for	Gaming, graphics rendering	Deep Learning, AI

Table2. Comparison of GPU and TPU on generic grounds.

In conclusion, when it comes to accelerating compute-intensive applications, GPUs and TPUs are two different kinds of specialized processors with various advantages and disadvantages. High memory capacity, high programmability, and relatively cheaper price are all characteristics of GPUs. They can be utilized for deep learning and other AI workloads in addition to the traditional uses of gaming and graphics rendering.

While deep learning models frequently use matrix-vector multiply-accumulate operations, TPUs are tuned for these tasks. They are less programmable than GPUs but offer better performance and less power usage. TPUs cost more, though, and they might not be able to handle larger models due to their reduced memory capacity.

It's crucial to consider the workload's specific requirements as well as the trade-offs between price, performance, and programmability when choosing between a GPU and a TPU. TPUs may significantly speed up large-scale deep learning models, even if GPUs are more adaptable and affordable. The decision between the two ultimately comes down to the particular requirements and limitations of the project.

III. Comparison of Intel Nervana Neural Processor and Google Tensor Processing Unit:

The final comparison used in conducting our research, to gain more insight is the comparison between the Intel Nervana Neural Processor and Google's Tensor Processing Unit.

The comparison can be represented in a table format as follows:

	Nervana Neural Processor (NNP)	**Tensor Processing Unit (TPU)**
Architecture	Neural network engine and tensor engine	Large number of Multiply-Accumulate units (MAC) units connected by high-bandwidth network
Performance	Fast for image and speech recognition	Effective for training large-scale language models
Memory	32MB on-chip memory	64GB HBM
Power Consumption	TDP of 180W	TDP of 250W
Availability	Only available in Intel's Nervana AI system	Available on Google Cloud Platform

Table3. Comparison of NNP and TPU.

To summarize, both NNP and TPU are both domain-specific processors that are well-suited for AI tasks overall. The NNP's innovative architecture combines a neural network engine and tensor engine for superior performance in image and speech recognition, but the TPU's conventional architecture with a sizable number of MAC units connected by a high-bandwidth network makes it especially effective in training large-scale language models. The NNP contains 32MB of on-chip memory, but the TPU has 64GB HBM. The NNP has a TDP of 180W, while the TPU has a TDP of 250W. Both CPUs were built with power efficiency in mind.

The TPU is available on Google Cloud Platform, where users can rent TPU resources to speed up their AI workloads. The NNP is presently only available in Intel's Nervana AI system. The decision between the NNP and TPU ultimately comes down to the exact workload requirements and the trade-offs between performance, power usage, and availability.

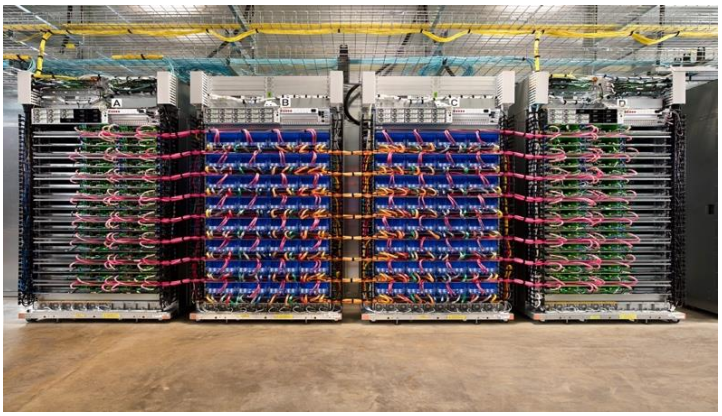

Figure 7. Google Tensor Processing Unit

IV. FUTURE PLANS

The creation of domain-specific processors for AI applications is a current subject of research because the field of AI is developing quickly. The following are some potential future directions for these processors:

- More specialized processing: Although current domain-specific processors are well suited for AI applications, more specialized processing is still possible. Future processors might be created expressly for jobs like speech, image, or natural language processing, which could greatly enhance performance.

- Greater flexibility is required to accommodate a variety of AI workloads, despite the fact that domain-specific processors are made for certain tasks. Future processors might be built to handle a variety of AI tasks, making it possible to process different workloads more effectively.

- Developments in the co-design of hardware and software: To attain optimum performance, future domain-specific processors will need both hardware and software modifications. To properly harness the potential of these computers, new software tools, programming languages, and algorithms will need to be created.

- Enhanced power efficiency: One of the main advantages of domain-specific processors is their superior performance to general-purpose processors in AI activities. To cut down on energy use and expenditures, additional advancements in power efficiency are still required.

- Integration with cloud computing: Domain-specific processors will need to be created for integration with cloud platforms as cloud computing for AI applications continues to expand. For seamless integration with cloud-based AI applications, this will necessitate the creation of new interfaces and APIs.

In general, the creation of domain-specific processors for AI applications is an intriguing field of study that is anticipated to result in major improvements in AI performance and capabilities. Future development of more potent, effective, and available AI applications may be made possible by ongoing research and development on these CPUs.

V. CONCLUSION

In conclusion, the creation of specialized processors for AI applications is a fascinating area of research. The field of AI hardware is quickly growing. When opposed to general-purpose processors, the adoption of AI processors offers a number of advantages in terms of cost, energy efficiency, and performance. The two most common types of AI processors right now are GPUs and TPUs, each of which has specific benefits and drawbacks.

Due to their adaptability and accessibility, GPUs are a popular choice for a variety of AI applications. They are suitable for both training and inference jobs and excel at handling parallelizable computations. Developers can adapt their AI models to match particular needs because to GPUs' high degree of flexibility and relatively simple programming. GPUs are less energy-efficient than other specialist processors because to their high operating costs and power requirements.

TPUs, on the other hand, are made expressly for AI workloads and are highly optimized for high-dimensional linear algebra operations such as matrix multiplication. They are a well-liked option for massive machine learning jobs because of their exceptional performance and energy efficiency. Thanks to Google's TensorFlow framework, which offers a high-level interface for creating and refining deep learning models, TPUs are also quite simple to utilize. TPUs are less suited to inference workloads than GPUs due to their lower flexibility and focus on training jobs.
Domain-specific processors have the advantage of using less energy when doing calculations than general-purpose processors. Because of this characteristic, they are perfect for mobile and edge devices where power consumption is a major limitation. In order to enable AI applications on devices with limited battery life, Qualcomm's AI engine, for example, is intended for mobile devices and is optimized for energy economy.

Other domain-specific processors for AI applications are also emerging in addition to GPUs and TPUs. The Nervana Neural Network Processor (NNP), for instance, was created by Intel specifically to handle the tasks associated with deep learning. The NNP has an innovative architecture that allows high-speed interconnects, allowing it to effectively manage huge data volumes. Additionally, the NNP features an adaptable programming model that enables AI developers.

In conclusion, the decision of an AI processor is based on the particular demands of the task and the trade-offs between flexibility, cost, and performance. TPUs provide good performance and energy efficiency for large-scale machine learning workloads, while GPUs are a versatile and widely available solution for a variety of AI applications. For particular AI workloads, other specialized processors, such the Nervana Engine, have distinct advantages. In the end, the creation of specialized processors for AI applications will make it possible to create AI apps that are more useful, approachable, and effective, with the potential to revolutionize entire sectors and enhance our daily lives.

REFERENCES

[1] “AI Drives Domain Specific Processors” by Yi Kang

link: https://ieeexplore.ieee.org/document/8579282/citations?tabFilter=papers#citations

[2] “A Semantic Future for AI” by R. Studer; A. Ankolekar; P. Hitzler; Y. Sure

link: https://ieeexplore.ieee.org/document/1667945

[3] “GCNAX: A Flexible and Energy-efficient Accelerator for Graph Convolutional Neural Networks” by Jiajun Li; Ahmed Louri; Avinash Karanth; Razvan Bunescu.

link: https://ieeexplore.ieee.org/document/9407104

[4] “FlexACC: A Programmable Accelerator with Application-Specific ISA for Flexible Deep Neural Network Inference” by En-Yu Yang; Tianyu Jia; David Brooks; Gu-Yeon Wei.

link: https://ieeexplore.ieee.org/document/9516466

[5] “Towards Automatic and Agile AI/ML Accelerator Design with End-to-End Synthesis” by Jeff Jun Zhang; Nicolas Bohm Agostini; Shihao Song; Cheng Tan.

link: https://ieeexplore.ieee.org/document/9516615