International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 2, Issue 2, PP. 10-17, 2021
ISSN: 3050-9416 | https://doi.org/10.63282/30509416/IJAIBDCMS-V2I2P102

Deep Learning Optimization in Cloud Environments: Challenges, Techniques, and Future Trends

Dr. Carlos Gutierrez,
University of Barcelona, AI & Computational Intelligence Lab, Spain.

Abstract: Deep learning in cloud environments offers benefits such as scalability, cost-effectiveness, and flexibility, yet presents several challenges. These include managing large datasets, allocating resources, and synchronizing processes across multiple machines. Training deep learning models on the cloud requires careful consideration of scalability, data storage and transfer, resource allocation, and model management. Overfitting and computation time also pose challenges, often addressed through regularization methods, dropout techniques, and optimization of training parameters. Techniques to optimize deep learning in the cloud include using GPUs and specialized deep learning processors like TPUs to speed up computation. Efficient data storage methods, such as those used with Neural Concept Shape, help manage large 3D numerical simulation files. Optimization strategies focus on improving computational efficiency and reducing costs. Deep learning applications in cloud environments span diverse fields, including image reconstruction, weather prediction, and financial mathematics. Future trends involve exploring hybrid forecasting models, investigating the impact of emerging technologies, and developing innovative methods for handling missing data to enhance predictive accuracy and robustness. The use of integrated photonic hardware accelerators and atomically thin semiconductors also shows promise for energy-efficient deep learning.

Keywords: Deep Learning, Cloud Computing, Optimization, Scalability, Resource Allocation, Future Trends

1. Introduction

Deep learning has revolutionized various fields, including image recognition, natural language processing, and predictive analytics. However, training complex deep learning models requires substantial computational resources and large datasets. Cloud computing offers a scalable and cost-effective solution for these demands, enabling researchers and practitioners to leverage powerful infrastructure without the burden of maintaining physical hardware.

1.1 The Rise of Deep Learning in the Cloud

The convergence of deep learning and cloud computing has opened up new possibilities for innovation. Cloud environments provide the necessary infrastructure, including virtual machines, GPUs, and specialized deep learning frameworks, to accelerate the development and deployment of deep learning models. The cloud's scalability allows for efficient processing of massive datasets, reducing training times and improving model accuracy. Furthermore, cloud-based deep learning platforms offer pre-trained models, APIs, and development tools that simplify the development process and make deep learning more accessible to a wider audience.

1.2 Challenges and Opportunities

Despite the advantages, deploying deep learning models in the cloud presents several challenges. These include data management, resource allocation, model optimization, and security concerns. Efficiently managing large datasets, minimizing data transfer costs, and ensuring data privacy are crucial for successful cloud-based deep learning. Optimizing model performance in the cloud requires careful consideration of hardware configurations, parallelization strategies, and distributed training techniques. Addressing these challenges will unlock the full potential of deep learning in the cloud, leading to new applications and advancements across diverse industries. The continuous evolution of cloud technologies and deep learning algorithms promises exciting opportunities for future research and development.

1.3. Framework and Architecture Overview

Cloud-based deep learning architectures represent a foundational strategy to harness the power of distributed computing for large-scale artificial intelligence tasks. The provided image highlights an upgraded layered structure that integrates various components essential for building, optimizing, and deploying deep learning applications in a cloud environment. This architecture ensures flexibility, scalability, and efficiency by dividing responsibilities into multiple layers, each catering to a specific aspect of the system. This design enables seamless transitions from raw data to meaningful inferences while accommodating the constraints and challenges of cloud resources.

The Application Layer represents the front-end applications directly impacting end-users. This layer includes tasks such as image and video recognition, click-through rate (CTR) estimation, natural language understanding, and speech recognition. These applications form the core value proposition of the system, showcasing the practical implementations of deep learning models trained and optimized in the cloud. By utilizing this layer, developers can align model capabilities with business needs or end-user requirements.

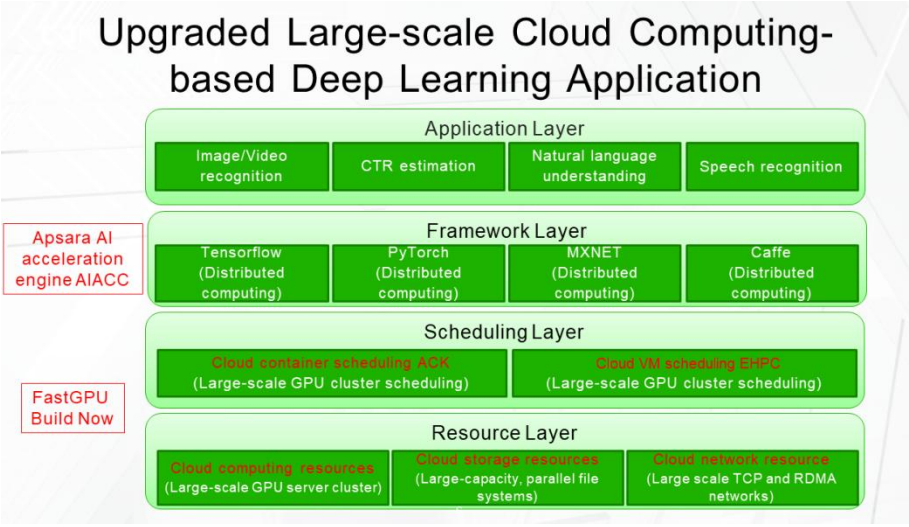

Figure 1: Large-Scale Cloud-Based Deep Learning Architecture

The Framework Layer underpins the application layer by providing the essential distributed computing frameworks necessary for model training and development. Popular frameworks such as TensorFlow, PyTorch, MXNet, and Caffe are included to support various types of models and optimization techniques. Each framework ensures the efficient distribution of training across multiple cloud resources, reducing time and improving performance. The addition of the Apsara AI acceleration engine (AIACC) highlights the use of specialized hardware and software optimizations that further enhance training speed and efficiency.

The Scheduling Layer addresses the allocation and optimization of computational resources. It ensures that large-scale GPU clusters and virtual machines (VMs) are effectively managed to meet the demands of deep learning workloads. Scheduling mechanisms like ACK and EHPC allow tasks to be prioritized based on resource availability, minimizing latency and maximizing GPU utilization. This layer forms the backbone of resource optimization, bridging the gap between the high-level frameworks and the underlying infrastructure.

At the foundation lies the Resource Layer, which provides the physical and virtual resources necessary for all the upper layers. These include GPU server clusters, large-scale storage resources with parallel file systems, and advanced network infrastructures like TCP and RDMA networks. These resources ensure the computational power, storage capacity, and high-speed communication required for real-time deep learning tasks.

2. Related Work

Research has been carried out to justify performing big data analytics on the cloud. Datasets in deep learning can be considered big data, involving large sets of images, videos, and audio files. A paper by Tsai, Chun-Wei, et al. focused on developing a high-performance platform to efficiently analyze big data and design an appropriate algorithm to find useful information. Salloum et al.'s paper studies analytics on the Apache Spark platform, a framework for big data analytics with an advanced in-memory programming model and upper-level libraries for scalable machine learning.

Work has been done to predict Virtual Machine (VM) workload in the cloud using deep learning, develop efficient mobile cloud systems for deep learning, and perform feature extraction for 3D point cloud data using autoencoders. Deep computation models have been designed to offload expensive operations to the cloud. Additional research demonstrates the benefits of hardware acceleration and high-performance gains in the cloud.

Deep learning and machine learning are effective in identifying and addressing cloud security threats. A study identified 4051 publications up to December 2023, highlighting key trend solutions such as anomaly detection, security automation, and the role of emerging technologies. Challenges such as data privacy, scalability, and explainability were also identified.

A technical analysis of using deep learning (DL) models, such as Recurrent Neural Networks (RNN), Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN), in cloud computing for accurate workload prediction has been performed. Other studies focus on developing deep convolutional neural networks (CNNs) to mine deep features of the cloud. These papers underscore the extensive research in performing analytics on the cloud and the various applications of big data analytics via the cloud.

3. Challenges in Deep Learning Optimization in Cloud Environments

Training deep learning models efficiently in cloud environments presents several significant challenges. These challenges span resource management, latency and bandwidth considerations, and data management issues. Overcoming these hurdles is crucial for leveraging the full potential of cloud computing in deep learning applications.

3.1. Resource Management

- **Computational Resource Constraints**: Deep learning models often require substantial computational resources, including CPUs, GPUs, and memory. Cloud environments offer various hardware configurations, but allocating the appropriate resources to ensure efficient training while avoiding over-allocation and unnecessary costs can be challenging. The increasing complexity of models and the growing size of datasets exacerbate this issue. Modern GPUs can handle complex, physics-based deep-learning challenges and refined geometries, making them essential for many advanced applications.
- **Scalability Issues**: Scaling deep learning training to accommodate large datasets and complex models presents a significant challenge. While cloud environments offer scalability, effectively distributing the workload across multiple machines requires careful coordination and optimization. Issues such as synchronization and parallelization must be addressed to maximize resource utilization and minimize training times.

3.2. Latency and Bandwidth

Communication Overhead in Distributed Training: Distributed training, where models are trained across multiple machines, is often necessary to handle large datasets and complex models. However, this approach introduces communication overhead due to the need to exchange data and synchronize updates between machines. Network latency and bandwidth limitations can significantly impact the efficiency of distributed training, leading to slower convergence and increased costs. Optimizing communication strategies and minimizing data transfer are crucial for mitigating these issues.

3.3. Data Management

- **Storage and Preprocessing of Large Datasets**: Deep learning models require large amounts of high-quality data to achieve optimal performance. Storing and transferring these datasets in cloud environments can be a significant challenge. The data transfer process can be slow and costly, and it can be difficult to ensure the security and privacy of the data. Efficient data storage solutions and optimized data preprocessing pipelines are essential for addressing these challenges. Neural Concept, for example, uses 3D numerical simulations as input to train deep learning models, which can result in very large files. Storing these files can become an issue, necessitating efficient storage solutions.
- **Data Privacy and Security**: Data privacy and security are paramount when dealing with sensitive information in cloud environments. Ensuring compliance with data protection regulations and implementing robust security measures to prevent unauthorized access are critical. Challenges associated with integrating deep learning into cloud security include data privacy concerns. Techniques such as encryption, anonymization, and secure data transfer protocols must be employed to protect data privacy and maintain the integrity of deep learning models.

3.4. Energy Efficiency

The high energy consumption of large-scale deep learning training is a growing concern. As machine learning (ML) technologies rapidly advance, their power consumption across various systems, from IoT devices to data centers, is surging. Benchmarking the energy efficiency of these systems is crucial for optimization. Studies show that faster programming languages tend to consume less energy; for example, compiled languages like C are typically more efficient than interpreted languages like Python because they translate directly into machine instructions. As models increase in complexity, the time to train becomes a critical factor, with researchers developing parallelization strategies to reduce training time by leveraging additional computational resources. However, this pursuit of faster training introduces complexity regarding energy

consumption at scale1. MLPerf Power benchmarking reveals the intricate relationship between system scale, training time, and energy consumption. Moreover, as newer and more complex generative models consume orders of magnitude more energy per inference, it is increasingly important to consider energy efficiency as a key metric in system design and optimization.

3.5. Fault Tolerance and Resilience

Handling failures in cloud infrastructures is a critical challenge for deep learning applications. Cloud environments are susceptible to various types of failures, including hardware malfunctions, network outages, and software bugs. These failures can interrupt the training process, leading to wasted resources and increased development time. Ensuring fault tolerance and resilience requires implementing mechanisms to detect and recover from failures automatically. Strategies include using redundant resources, implementing checkpointing and rollback mechanisms, and employing distributed training techniques that can tolerate node failures. An energy-efficient strategy for big data in a cloud environment could use deep reinforcement learning. By implementing these strategies, deep learning applications can continue to operate reliably even in the face of failures, ensuring the successful completion of training tasks.

3.6. Other Technical and Ethical Challenges

Besides the aforementioned technical challenges, cost considerations, vendor lock-in, and ethical concerns also pose significant hurdles in deep learning optimization within cloud environments. The cost of cloud resources, including compute instances, storage, and data transfer, can be substantial, especially for large-scale deep learning projects. It is crucial to optimize resource utilization and explore cost-effective alternatives, such as spot instances or serverless computing, to minimize expenses. Vendor lock-in is another concern, as organizations may become dependent on specific cloud providers and their proprietary technologies, making it difficult to switch providers or adopt new technologies. Addressing these challenges requires careful planning, strategic decision-making, and a commitment to ethical principles to ensure that deep learning applications are developed and deployed responsibly.

Deep learning optimization within cloud environments, showcasing the interaction between various components of the deep learning system and the cloud infrastructure. It provides a layered approach to understanding the workflows, ranging from the definition of training goals to the deployment of inference-ready models. This architecture highlights the collaborative roles of different actors, including data scientists, cloud administrators, and end-users, each interacting with distinct system modules.

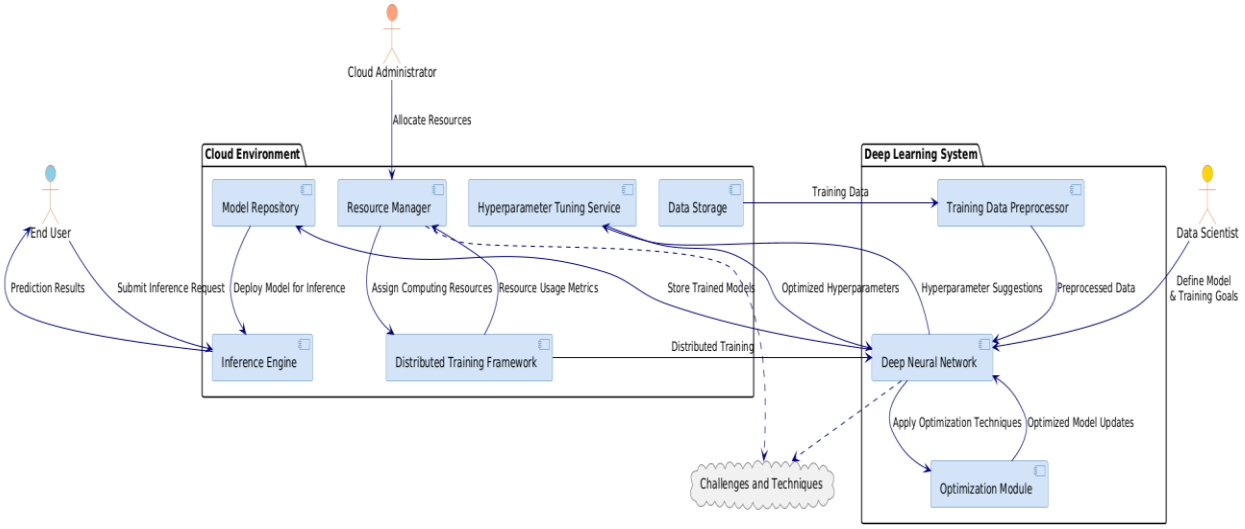

Figure 2: Deep Learning Optimization Architecture in Cloud Environments

The Deep Learning System comprises modules like the training data preprocessor, optimization module, and the core deep neural network. These components work together to process raw data, optimize training workflows, and iteratively refine the deep learning models. Data scientists define the model structure and training goals, which are processed by the system to ensure efficient training and optimization. The optimization module applies advanced techniques to improve model accuracy and efficiency, creating a feedback loop that continuously updates the neural network. the Cloud Environment forms the backbone of the architecture, offering scalable resources and essential services for distributed deep learning. Components like the resource manager, data storage, hyperparameter tuning service, and distributed training framework collectively ensure that the computational and storage demands of deep learning are met. The resource manager dynamically allocates cloud resources

based on workload requirements, while the distributed training framework divides the workload across multiple nodes to accelerate training.

The interdependence between the two systems, with bidirectional data flows linking the cloud environment to the deep learning system. For instance, training data from the cloud's storage module is preprocessed and used by the deep neural network, while trained models are stored in the cloud-based model repository for future deployment. Similarly, hyperparameter tuning services provide optimized configurations that are fed back into the training process to enhance model performance.

A critical aspect depicted in the diagram is the interaction with external actors. Cloud administrators oversee the allocation of resources within the cloud environment, ensuring that workloads are efficiently managed. End-users, on the other hand, interface with the system by submitting inference requests to the deployed model via the inference engine. The system's ability to process these requests and return predictions underscores its real-world applicability.

The cloud-based design of the system enables it to address key challenges such as resource efficiency, scalability, and latency. By integrating optimization techniques, distributed training, and hyperparameter tuning, the architecture represents a robust solution for deploying and managing deep learning models in cloud environments. This framework serves as a blueprint for addressing the complexities of large-scale AI applications while maintaining efficiency and adaptability.

4. Techniques for Optimizing Deep Learning Models in Cloud Environments

Optimizing deep learning models in cloud environments is essential for achieving high efficiency and performance while minimizing computational costs. A wide range of techniques, spanning model optimization, cloud-specific strategies, resource scheduling, and hyperparameter tuning, can be employed to address the challenges posed by large-scale deep learning tasks. These methods ensure that the training and deployment of models remain cost-effective, scalable, and adaptable to dynamic workloads.

4.1. Model Optimization Techniques

Model optimization plays a key role in reducing the computational and memory demands of deep learning models. Pruning is one such technique, where less critical neurons in the network are identified and removed to reduce the model size without significantly affecting its accuracy. This process often includes a fine-tuning step to regain any lost performance. Quantization is another widely used approach, reducing the numeric precision of model weights to lower memory usage and computation time, making it particularly useful for edge and mobile deployments. Knowledge distillation further enhances efficiency by transferring the learning of a complex "teacher" model to a simpler "student" model, allowing the latter to achieve similar accuracy while requiring fewer computational resources.

4.2. Cloud-Specific Techniques

The scalability and resource-sharing capabilities of cloud platforms open the door to specialized optimization strategies. Distributed training frameworks, such as TensorFlow and PyTorch, enable parallel training of deep learning models across multiple machines. This reduces training time and allows the system to handle larger datasets and complex architectures efficiently. Federated learning, on the other hand, facilitates decentralized model training directly on distributed devices, preserving data privacy while reducing the need for extensive data transfer to the cloud. Serverless computing and auto-scaling further enhance cloud efficiency by dynamically allocating resources on-demand. These strategies optimize cost and resource usage by scaling up during high workload periods and scaling down during idle times.

4.3. Resource Scheduling and Management

Efficient scheduling and resource management are critical for optimizing deep learning workloads in cloud environments. Advanced scheduling algorithms are employed to allocate resources effectively, ensuring optimal utilization and minimizing task completion times. Containerization technologies, such as Kubernetes, add another layer of efficiency by providing consistent environments for deploying and managing deep learning models. These containers ensure portability and scalability, allowing deep learning workflows to be seamlessly migrated across cloud platforms while maintaining consistency and performance.

4.4. Hyperparameter Optimization

Finding the optimal hyperparameters is crucial for maximizing the performance of deep learning models. Techniques like grid search, random search, and Bayesian optimization systematically explore the hyperparameter space to identify the best configurations. Tools like AI Platform Optimizer automate this process, reducing the time and effort required for experimentation. For more complex scenarios, hybrid approaches such as combining deep learning with optimization

techniques like Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) can be employed. For example, a hybrid CNN-LSTM model has been used to predict cloud resource utilization dynamically. The CNN component extracts intricate workload patterns, while the LSTM component captures temporal dependencies, providing accurate forecasts for future virtual machine workloads. This hybrid approach helps address load balancing and over-provisioning challenges by integrating multi-resource utilization predictions.

4.5. Holistic Impact of Optimization Techniques

By combining these strategies, organizations can build highly efficient deep learning pipelines tailored for cloud environments. Model optimization techniques reduce computational demand, while cloud-specific methods leverage the inherent advantages of distributed resources. Advanced scheduling and resource management ensure optimal infrastructure utilization, and hyperparameter optimization fine-tunes the performance of models. Together, these approaches create a robust framework for deploying deep learning models that are not only high-performing but also cost-effective and scalable for diverse applications.

4.6. Parallel and Asynchronous Training Methods

Parallel and asynchronous training techniques are critical for efficiently scaling deep learning models across multiple devices and nodes, particularly in large-scale cloud environments. These methods are designed to maximize the utilization of available hardware resources while reducing training times and computational bottlenecks. Two prominent approaches to parallel training are data parallelism and model parallelism, each tailored for specific scenarios based on dataset size and model complexity.

Data parallelism involves duplicating the model parameters across multiple GPUs and assigning different subsets of data to each GPU for simultaneous processing. This approach excels in scenarios where datasets are large, and models are of moderate size, as it allows GPUs to process their respective data batches independently. Frameworks like TensorFlow and PyTorch provide native support for data parallelism, enabling straightforward workload distribution across GPUs. However, a significant challenge with this approach lies in synchronizing and aggregating gradients from all devices, which can create communication overhead and slow down the training process if not managed effectively.

Model parallelism, on the other hand, divides the model itself across multiple GPUs. Different parts of the model, such as layers or groups of neurons, are assigned to separate devices. This method is particularly useful for training very large models that cannot fit into the memory of a single GPU, regardless of dataset size. Variants like pipeline parallelism enhance this approach by dividing the model into segments and processing different mini-batches concurrently on separate GPUs. While model parallelism addresses memory constraints, it introduces challenges in handling communication overhead, as intermediate outputs need to be transferred between devices frequently.

Asynchronous training is another technique that can significantly improve scalability and reduce synchronization costs. In this approach, each GPU or node trains independently and updates a shared model periodically rather than synchronizing after every step. This method is especially beneficial for large-scale optimization problems, where frequent synchronization can become a bottleneck. Asynchronous training frameworks, like Atom, facilitate decentralized training of vast models using cost-effective hardware. By allowing parallel actor-learners to work independently, asynchronous training can stabilize the overall training process and accelerate convergence in distributed settings.

4.7. Techniques to Reduce Latency and Bandwidth Issues

Latency and bandwidth constraints can severely impact the performance of distributed deep learning systems, particularly in cloud environments where communication overhead is a critical factor. To address these challenges, several techniques have been developed to minimize data transfer requirements and optimize synchronization processes during training.

One effective method is compression techniques for model updates, such as quantization, pruning, and knowledge distillation. By reducing the size of model parameters and updates, these techniques help alleviate bandwidth bottlenecks and accelerate synchronization in distributed training. For example, quantization reduces the precision of weights and activations, thereby decreasing the size of transmitted data. Similarly, pruning eliminates less important model components, further shrinking the model's footprint and making updates more efficient.

Another impactful approach is gradient sparsification, which selectively transmits only the most critical gradients during the synchronization step. By reducing the volume of data communicated between devices, gradient sparsification minimizes communication overhead and improves overall training speed. This technique is especially valuable in distributed systems

where network latency and bandwidth limitations can otherwise dominate the training process. Additionally, it allows deep learning models to maintain their performance while significantly cutting down on resource usage.

5. Case Studies and Applications

AI and cloud computing have been integrated for scalable data processing across various sectors. For example, in e-commerce, AI-powered cloud data processing pipelines enable real-time customer behavior analysis to optimize product recommendations. A case study showed the system processed 10 million user interactions daily, achieving an average latency of 2.5 seconds for generating recommendations and a model accuracy of 85%3. The system also demonstrated scalability, with a 10% increase in traffic leading to only a 5% increase in processing time.

In financial services, AI effectively detects fraudulent activities3. A case study focused on fraud detection processed millions of transactions, achieving higher accuracy (92%) compared to e-commerce. The system also scaled efficiently, though the operational costs were higher due to the complexity of the fraud detection model and the need for real-time processing.

Neural Concept has been active in building scalable frameworks to train deep learning models efficiently. With Neural Concept Shape (NCS), 3D numerical simulations can be used as input to train deep learning models. Efficient data storage methods, such as those used by Neural Concept with FUSE libraries, allow users to train their models directly from secure cloud storage without impacting computation speed. They also use the cache functionality of TensorFlow data API, caching the dataset to a local SSD disk during training to enable very efficient data retrieval.

A deep learning-driven Max-out prediction model can efficiently forecast future workload by providing a balanced approach for enhanced scheduling with the Tasmanian Devil-Bald Eagle Search (TDBES) optimization algorithm. The results obtained proved that the TDBES scored efficacy in makespan with 16.75%, migration cost with 14.78%, and a migration efficiency rate of 9.36% over other existing techniques like DBOA, WACO, and MPSO1.

6. Future Trends and Research Directions

- **Integration of AI and ML in Cloud Services**: The use of artificial intelligence and machine learning is expected to rise, aiding organizations in various ways. Some cloud providers already offer AI and ML services, and this is expected to gain even more traction. Integrating AI and ML in cloud computing will reduce costs and enhance the effectiveness of cloud services. This synergy leads to more sophisticated, efficient, and cost-effective solutions. AI's introduction will help cloud providers automate redundant processes like data management and improve resource efficiency. AI will also help in predicting trends, identifying patterns, and detecting anomalies, which will improve cloud computing in many ways.
- **Edge Computing**: As the use of cloud-based services increases, edge computing will also gain popularity. Edge computing helps organizations reduce downtime by processing data in a local system, preventing issues associated with the cloud, enhancing data security, and saving capital. The future will see a massive surge in edge computing due to its efficient and improved data processing power. As the requirement for real-time data analytics increases, organizations will start leveraging edge computing. Edge AI, which applies AI closer to the source of data, improves the efficiency of real-time decisions.
- **Enhanced Security Measures**: With the increasing use of cloud computing, cyber-attacks are also expected to rise. Cloud providers are expected to leverage advanced technology along with AI and ML to come up with advanced security measures like access control, encryption, and threat detection. Cloud security will become intelligent, automated, and reliable, driven by advances in AI, machine learning, and quantum computing.
- **Quantum Computing**: Quantum computing has the potential to revolutionize machine learning by exponentially increasing computational power. Quantum algorithms can solve problems that are currently intractable for classical computers. Integration with ML is likely to accelerate more innovation across varied industries, though quantum computing is still in its early stages.
- **Cloud-Native Development Approach**: Businesses will start adopting cloud-native development approaches, utilizing microservices and containers to gain agility, efficiency, and scalability in the development process. Many top organizations are already promoting cloud-native development.

7. Conclusion

Deep learning optimization in cloud environments presents a multifaceted challenge requiring the careful consideration of resource management, latency, data handling, energy efficiency, and ethical concerns. Techniques such as model optimization, distributed training, and optimized resource scheduling play a vital role in improving performance and cost-effectiveness. As

we've explored, each of these areas has its own complexities and trade-offs, necessitating a tailored approach for each specific application and infrastructure.

The trends of AI integration, edge computing, enhanced security, quantum computing, and cloud-native development promise to reshape the landscape of deep learning in the cloud. Continued research and innovation in these areas will be crucial to unlock the full potential of cloud-based deep learning, enabling new breakthroughs and applications across a wide range of domains. By addressing the challenges and embracing emerging trends, we can pave the way for a future where deep learning in the cloud is more accessible, efficient, and impactful than ever before.

References

1. Dean, J., Patterson, D., & Young, C. (2018). "A new golden age in computer architecture: Empowering the machine-learning revolution." IEEE Micro, 38(2), 21-29.
2. Li, Y., & Wu, J. (2020). "A survey on federated learning: Concept, applications, and future directions." IEEE Transactions on Artificial Intelligence, 1(1), 49-76.
3. Zhang, C., et al. (2019). "A unified optimization framework for deep neural networks in cloud environments." IEEE Transactions on Cloud Computing, 7(3), 690-703.
4. Abadi, M., Barham, P., Chen, J., et al. (2016). "TensorFlow: A system for large-scale machine learning." OSDI 2016.
5. Wang, S., Tuor, T., Salonidis, T., et al. (2018). "Adaptive federated learning in resource-constrained edge computing systems." IEEE Journal on Selected Areas in Communications, 37(6), 1205-1221.
6. Chen, J., Lin, X., Kang, J., & Yu, H. (2020). "Optimizing deep learning models for edge-cloud computing systems." IEEE Transactions on Cloud Computing, 8(3), 756-769.
7. Park, J., Samarakoon, S., Bennis, M., & Debbah, M. (2019). "Wireless network intelligence at the edge." Proceedings of the IEEE, 107(11), 2204-2239.
8. Gholami, A., Kim, S., Dong, Z., et al. (2021). "A survey on quantization techniques for deep learning." arXiv preprint arXiv:2103.13630.
9. Li, H., Ota, K., & Dong, M. (2018). "Learning IoT in edge: Deep learning for the Internet of Things with edge computing." IEEE Network, 32(1), 96-101.
10. Zhu, H., Huang, C., Zhang, R., & Larsson, E. G. (2020). "Broadband analog aggregation for low-latency federated edge learning." IEEE Transactions on Wireless Communications, 19(1), 491-506.