**University of Central Missouri**

**Department of Computer Science & Cybersecurity**

**CS5720 Neural network and Deep learning**

**Spring 2025**

**Home Assignment 2. (Cover Ch 4,5)**

**Student name:**

**Pothuru.Sivarkamani Naga Mruthyunjaya**

**Submission Requirements:**

* Total Points: 100
* Once finished your assignment push your source code to your repo (GitHub) and explain the work through the ReadMe file properly. Make sure you add your student info in the ReadMe file.
* Submit your GitHub link and video on the BB.
* Comment your code appropriately ***IMPORTANT.***
* Make a simple video about 2 to 3 minutes which includes demonstration of your home assignment and explanation of code snippets.
* Any submission after provided deadline is considered as a late submission.

**Question 1: Cloud Computing for Deep Learning (20 points)**

Cloud computing offers significant advantages for deep learning applications.

(a) Define **elasticity** and **scalability** in the context of cloud computing for deep learning. (10 points)  
(b) Compare **AWS SageMaker**, **Google Vertex AI**, and **Microsoft Azure Machine Learning Studio** in terms of their deep learning capabilities. (10 points)

**Expected Output**

Write the definition and comparison for (a) and (b). No code needed.

**Answer:**

**a**

**Elasticity** refers to the ability of a cloud computing system to automatically allocate and deallocate computing resources based on the current demand. For deep learning applications, elasticity is crucial because the computational requirements can vary significantly. During model training, there may be periods of high demand for computational power, while at other times, less power may be required. Cloud services that are elastic can scale resources up and down dynamically to optimize costs and performance. For example, when training a deep learning model, a cloud service can quickly provision additional resources (e.g., GPU instances) as the workload increases and reduce resources once the demand decreases.

**Scalability** refers to the capability of a cloud system to handle an increasing workload or expand its resources to meet the growing demands. In deep learning, scalability is essential when training large models or working with vast amounts of data. A cloud environment that is scalable can efficiently grow by adding more resources, such as storage, processing power, or GPUs, to accommodate larger datasets, more complex models, or a greater number of concurrent tasks. Cloud platforms for deep learning should support both **horizontal scalability** (adding more machines) and **vertical scalability** (adding more power to a single machine).

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Scalability** | **Elasticity** |
| **Definition** | Ability to handle increased workload by adding resources. | Dynamic adjustment of resources based on demand fluctuations. |
| **Resource Allocation** | Resources are added or removed manually in response to workload changes. | Resources are automatically scaled up or down based on demand. |
| **Timing** | Scaling usually occurs reactively in response to increased demand. | Resources are scaled dynamically in real-time as demand changes. |
| **Management** | Typically requires human intervention for resource adjustments. | Resource allocation is automated, minimizing manual involvement. |
| **Flexibility** | Offers flexibility in scaling but may not be instantaneous. | Offers instant and automated resource adjustments as needed. |
| **Use Cases** | Commonly used in systems where workload patterns are predictable. | Ideal for applications with highly variable or unpredictable workloads. |
| **Cost Implications** | May lead to over-provisioning during low-demand periods. | Enables cost optimization by scaling resources precisely to demand. |
| **Response Time** | Scaling operations may take time to implement and may cause downtime during adjustments. | Resource adjustments are instantaneous, minimizing downtime and ensuring seamless performance. |
| **Complexity** | Generally simpler to implement and manage, especially in environments with stable workloads. | Requires sophisticated automation and monitoring systems to dynamically adjust resources based on demand. |
| **Efficiency** | May result in underutilization of resources during low-demand periods. | Optimizes resource utilization by scaling resources precisely to match demand, thus reducing waste. |

**(b)**

**Comparison:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **AWS SageMaker** | **Google Vertex AI** | **Microsoft Azure ML Studio** |
| **Supported Frameworks** | TensorFlow, PyTorch, MXNet, Chainer | TensorFlow, PyTorch, scikit-learn, custom | TensorFlow, PyTorch, MXNet, custom |
| **Pre-built Containers** | Yes | Yes | Yes |
| **Training Infrastructure** | Elastic infrastructure with auto-scaling, GPU, P3, P4 instances | TPUs, GPUs, distributed training | GPU-based VMs, distributed training |
| **AutoML** | No | Yes (AutoML capabilities) | Yes (AutoML and Designer) |
| **Hardware Acceleration** | EC2 P3/P4 GPUs | TPUs, GPUs | Azure ND-series GPUs |
| **Scalability** | High (Elastic scaling, Spot instances) | High (TPUs, GPUs, multi-node configurations) | High (distributed training, AKS) |
| **MLOps / Automation** | SageMaker Pipelines | Vertex AI Pipelines | Azure ML Pipelines |
| **Model Monitoring** | Yes (Explainability, Drift detection) | Yes (Drift detection, performance tracking) | Yes (drift detection, monitoring) |
| **Ease of Use** | Studio with Jupyter notebooks | Vertex AI Workbench with Jupyter notebooks | Studio with drag-and-drop interface |

**Conclusion:**

* **AWS SageMaker** is a robust and highly scalable solution with a wide variety of services for deep learning, offering strong integration with the broader AWS ecosystem and flexible infrastructure options for both training and deployment.
* **Google Vertex AI** stands out with its **TPU** support, which is ideal for deep learning tasks requiring significant computational power, and its user-friendly **AutoML** capabilities.
* **Microsoft Azure ML Studio** offers a highly integrated ecosystem for enterprises and advanced users, with powerful **drag-and-drop** capabilities and strong integration with Microsoft services, making it easy to build, train, and deploy models at scale.