Adaptation of Large Foundation Models

About This Whitepaper

Tuning a large model involves adjusting and adapting a pre-trained model to perform specific tasks with the user provided training dataset. This white paper is a technical reference aimed at outlining Google's approach on adapter tuning. Developers who are responsible for deploying machine learning models, product managers, business leaders, and security engineers will find parts of this white paper relevant, particularly around how data is handled in various stages and the current limitations. Implementation details are current as of July 11th, 2023 and are subject to change.

Overview

Background on Foundation Models

Foundation models, also known as large-scale pre-trained models, have become a transformative force in the field of artificial intelligence (AI) and machine learning (ML). These models are typically trained on vast amounts of diverse data, allowing them to learn general patterns and representations that can be applied across various domains and tasks.

The power of foundation models lies in their ability to learn high-quality, transferable features from the data they are trained on during the pre-training phase. This pre-training phase enables the models to capture relationships and develop an understanding of the data, which can then be fine-tuned to specific tasks. As a result, foundation models have proven to be highly effective in numerous applications, such as natural language processing, computer vision, and reinforcement learning.

The widespread success of foundation models can be attributed to several factors, including:

- **Large-scale data**: Foundation models are trained on large datasets, which provide a source of diverse information for learning general patterns and structures. This data exposure allows the models to develop a nuanced understanding of a wide variety of domains and tasks.
- **Advanced architecture**: The architecture of foundation models, such as the [Transformer](), enables them to capture complex relationships.
- **Hardware advancements:** The performance of foundation models tends to improve with increased size and computational resources. Researchers and organizations have been

able to push the boundaries of these models by scaling them up, leading to even more powerful and capable systems. This scalability enables foundation models to tackle increasingly complex tasks and achieve state-of-the-art performance across various domains. New generations of accelerators (e.g. TPUs, GPUs) provide better computation efficiency, lower energy consumption, higher memory size and bandwidth per chip, and better connectivity that allows the models to scale up to the current sizes.

Adapter Tuning

Parameter Efficient Fine Tuning (PEFT) is a class of fine-tuning methods that help foundational models adapt to specialized tasks by incorporating domain-specific or organization-internal data. Through the addition of a minimal number of parameters (adapters), these techniques enable efficient model modification and alignment with the target domain.

These techniques enable tuning of the model to specific tasks without having to rebuild the entire foundation model. Shared deployment of a foundation model can be quickly augmented with adapter weights that are specific to a particular task or domain at runtime. This allows multiple users of the same foundation model to privately augment and manage the same foundation model with training that is unique to their needs or deployment requirements.

Security and Privacy Principles

In this section, we go through the high-level security and privacy principles that guide our design process for tuning and serving Foundation Models with respect to Google Cloud and our customers' data.

Customer data (e.g. prompts, input training data, etc.) will not be logged or used for improving the Google foundation models without the customer's permission.

Input data such as prompts are customer data and stored securely at every step along the way - encrypted at rest and in transit. Adapter models are also stored securely, and Customers will have sole access to use any adapter models.

Customers will be able to control the encryption of stored adapters by using customer-managed encryption keys (CMEK), and delete adapters at any time.[1]

---

[1]CMEK will be available at GA

Design of Adapter Tuning on Vertex AI

Our Parameter Efficient Fine Tuning jobs run on Cloud TPUs/GPUs through Vertex AI's training service. Vertex AI creates a separate *tenant project* for each customer project and runs the training workloads on Compute Engine VMs on the tenant project.

During the Parameter Efficient Fine Tuning process, the associated VMs in the tenant project load the frozen model weights and the training dataset that consists of hundreds of prompt/response examples. The adapter weights are trained by using gradient-based optimization methods, and then checkpointed to a per-customer bucket in the tenant project during the training. Once the training is done, the final adapter weights are stored in the same per-customer bucket for deployment.
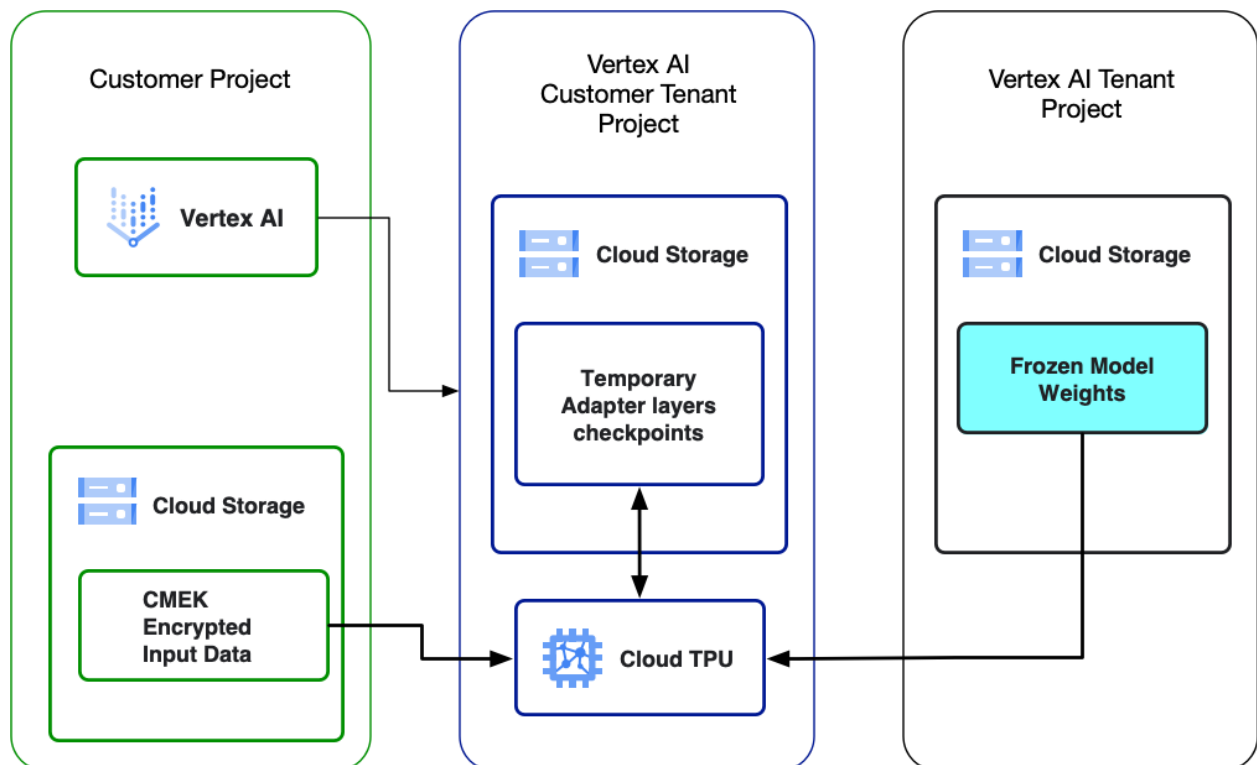
Security considerations

Throughout the training process, user input data remains confined to the user's storage bucket in the customer project, ensuring no external data transfer. Data can be encrypted by using

Customer-Managed Encryption Keys (CMEK)[2] on Cloud Storage and accessed via the user-provided service account for enhanced security and inside the VPC security perimeter of the customer.

Temporary checkpoints generated in the tenant's project bucket are subject to automatic deletion within a 30-day period, minimizing the risk of unauthorized access or unintended data retention.

Access to the tenant project by anyone in Google other than the customer, requires a valid business justification, such as a request for Google to assist in debugging an issue or a system outage. Any request is logged through Access Transparency logs to maintain a comprehensive record of all interactions and safeguard against potential security breaches.

Design of Adapter serving on Vertex AI

Once the adapter layers are trained, the weights are uploaded as a model to a bucket in the Vertex Prediction per-customer tenant project and is deployed to an Endpoint in the customer's project. When serving a request on an endpoint, the adapter weights are loaded from the bucket, cached to minimize latency and sent to the foundation model along with the original request (e.g. user's prompt) for inference. The latency overhead of transferring the adapter layers is typically negligible compared to the inference of a large model.

Foundation model service has the frozen model weights loaded during the start up. It receives the adapter weights for the duration of an inference, runs through the request and returns the results, without modifying the model or storing the request.

---

[2]CMEK, User Provided Service accounts and VPC Service controls are currently not supported.
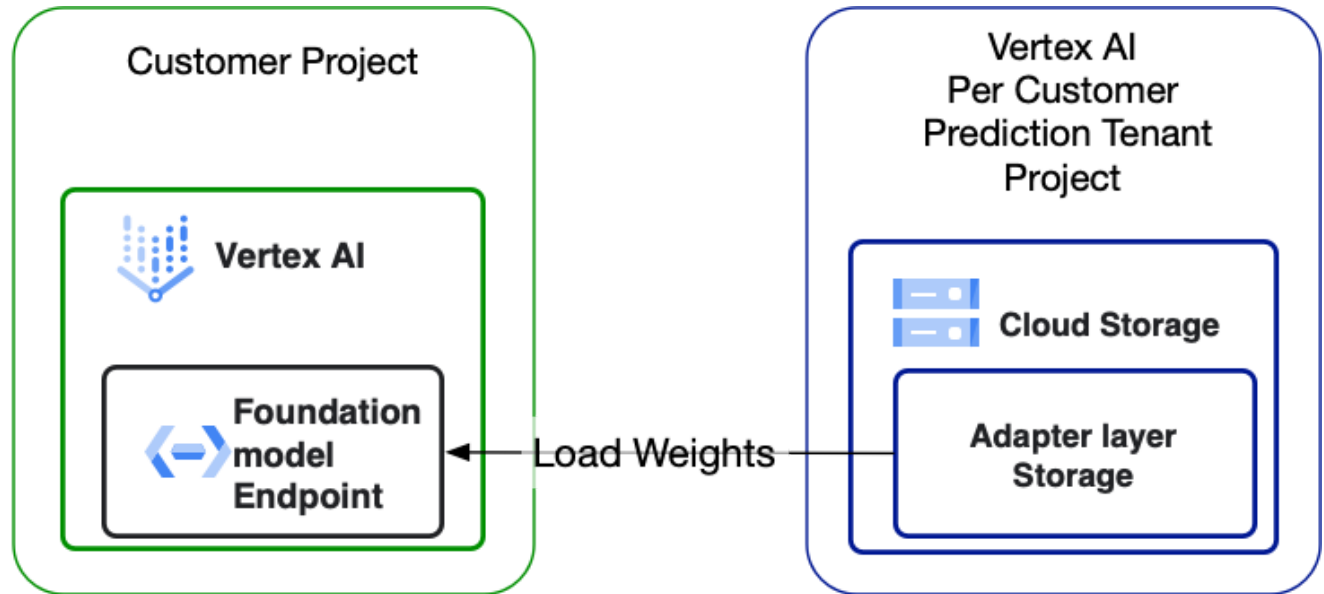
Security considerations

The adapter layer is stored on Cloud Storage with the option of using CMEK[3], Access Transparency with audit logging, and multi-party authentication for many of the administrative processes.

You can use VPC Service Controls (VPC-SC) to create a service perimeter that protects the endpoint. The endpoint can be accessed only from within the VPC service perimeter and access can be controlled through IAM permissions.

---

[3] CMEK and Access Transparency will be available during General Availability of this feature.