1. What does a SavedModel contain? How do you inspect its content?

**SavedModel Content:**

* A SavedModel contains the TensorFlow model's architecture, weights, and computation graph.
* It includes model-specific metadata like signatures for input and output tensors, which is useful for serving.
* You can inspect its content using TensorFlow's tools like saved\_model\_cli, which allows you to explore the graph and signatures.

1. When should you use TF Serving? What are its main features? What are some tools you can use to deploy it?

**Using TF Serving:**

* You should use TensorFlow Serving (TF Serving) when you want to deploy machine learning models in a production environment for serving predictions.
* Main Features: Versioning, Load Balancing, Scaling, REST and gRPC APIs, Support for Multiple Models, GPU and TPU Acceleration.
* Tools: TensorFlow ModelServer, Docker for containerization, Kubernetes for orchestration.

1. How do you deploy a model across multiple TF Serving instances?

**Deploying Across Multiple TF Serving Instances:**

* Deploying across multiple instances involves setting up TF Serving instances on different servers and configuring a load balancer to distribute requests.
* The load balancer distributes incoming requests to different instances, achieving load distribution and fault tolerance.

1. When should you use the gRPC API rather than the REST API to query a model served by TF Serving?

**gRPC vs. REST API in TF Serving:**

* gRPC is preferred when low latency and high performance are critical.
* REST API is more widely supported and easier to use in web applications.
* Choose based on your application's requirements and constraints.

1. What are the different ways TFLite reduces a model’s size to make it run on a mobile or embedded device?

**Reducing Model Size with TFLite:**

* TFLite reduces model size for mobile/embedded devices through quantization (8-bit integer quantization), model pruning, and weight quantization.
* Post-training quantization, quantization-aware training, and sparsity optimization techniques can be applied.

1. What is quantization-aware training, and why would you need it?

**Quantization-Aware Training:**

* Quantization-aware training is a training technique that simulates the effects of quantization during training.
* It helps the model to become robust to the quantization process, resulting in smaller model sizes without significant loss of accuracy.

1. What are model parallelism and data parallelism? Why is the latter generally recommended?

**Model Parallelism vs. Data Parallelism:**

* **Model Parallelism:** Splits the model into parts and runs each part on a different device/server. Typically used for very large models that don't fit in a single device's memory.
* **Data Parallelism:** Replicates the model on multiple devices/servers and divides the data into batches. Each replica computes gradients for its batch and averages them. Generally recommended as it's more efficient for most scenarios.

1. When training a model across multiple servers, what distribution strategies can you use? How do you choose which one to use?

**Distribution Strategies for Training:**

* Distribution strategies include MirroredStrategy (multi-GPU), TPUStrategy (Google TPUs), and others.
* Choice depends on available hardware and the scale of your training task.
* For multi-GPU training, MirroredStrategy is commonly used as it allows synchronous training on multiple GPUs.