The provided text is a detailed summary about training on TPU (Tensor Processing Unit) with TensorFlow. Here are the key points covered in the summary:

TPU Overview: TPUs are hardware designed by Google to accelerate tensor computations in neural networks, similar to GPUs. They can be used for both training and inference. TPUs are typically accessed through Google's cloud services, but smaller TPUs can also be accessed directly for free through platforms like Google Colab and Kaggle Kernels.

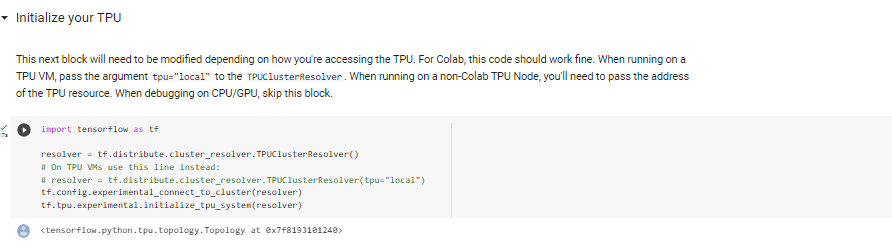
Types of TPUs: There are two ways to access TPUs: TPU Nodes and TPU VMs. TPU Nodes involve accessing TPUs indirectly via a remote node, while TPU VMs allow for direct connection to the machine with the attached TPU. TPU VMs are generally easier to work with, especially when it comes to data pipeline setup.

TPU Sizes: A single TPU can run 8 replicas, and TPUs are organized in pods that can run hundreds or thousands of replicas simultaneously. When using multiple TPUs but less than a whole pod, it is referred to as a pod slice. Free TPUs accessed through Colab usually provide a single v2-8 TPU.

XLA and TPUs: XLA (Accelerated Linear Algebra) is an optimizing compiler used by TensorFlow and JAX. XLA is mandatory when training on TPUs. Enabling XLA can improve performance, and its compatibility can be tested on CPUs or GPUs before running on TPUs. However, XLA compilation should be removed when training on TPUs, as it happens implicitly.

Making Models XLA Compatible: There are three core rules for XLA compatibility: avoiding data-dependent conditionals, data-dependent shapes, and recompilation for different input shapes. Adhering to these rules ensures XLA compatibility. Strategies like padding can be used to handle variable input shapes.

Training on TPU: Once the code is XLA-compatible and the dataset is prepared, training on TPUs involves initializing the TPU, creating the model and dataset within a TPUStrategy scope, and running the training using model.fit(). The TPU example notebook provides code samples and guidance for training on TPUs.



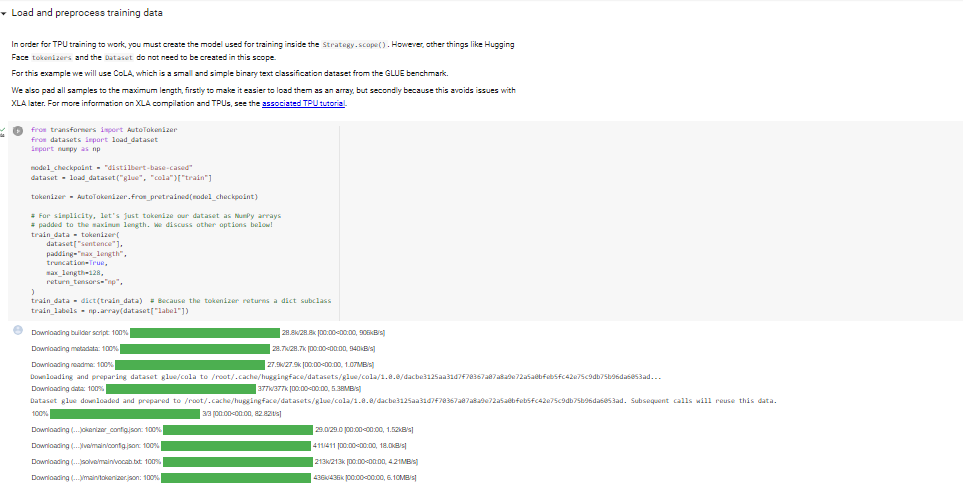
As mentioned earlier, after the code is XLA-compatible we must create our model using TPU Strategy

A picture containing text, screenshot, font

Description automatically generated

A picture containing text, screenshot, font

Description automatically generated



A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with low confidence

