This tutorial shows you how to train the Transformer (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py) model (from *Attention Is All You Need* (https://arxiv.org/abs/1706.03762)) with Tensor2Tensor (https://github.com/tensorflow/tensor2tensor) on a Cloud TPU.

The Transformer model uses stacks of self-attention layers and feed-forward layers to process sequential input like text. It supports the following variants:

- `transformer` (decoder-only) for single sequence modeling. Example use case: language modeling.
- `transformer` (encoder-decoder) for sequence to sequence modeling. Example use case: translation.
- `transformer_encoder` (encoder-only) runs only the encoder for sequence to class modeling. Example use case: sentiment classification.

The Transformer is just one of the models in the Tensor2Tensor (https://github.com/tensorflow/tensor2tensor) library. Tensor2Tensor (**T2T**) is a library of deep learning models and datasets as well as a set of scripts that allow you to train the models and to download and prepare the data.

If you plan to train on a TPU Pod slice, please make sure you go over this document (/tpu/docs/training-on-tpu-pods) that explains the special considerations when training on a Pod slice.

Before starting this tutorial, follow the steps below to check that your Google Cloud project is correctly set up.

1. Sign in (https://accounts.google.com/Login) to your Google Account.

   If you don't already have one, sign up for a new account (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note**: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

Go to the project selector page (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. Learn how to confirm billing is enabled for your project (/billing/docs/how-to/modify-project).
4. Verify (/tpu/docs/quota#requesting_additional_quota) that you have sufficient quota to use either TPU devices or Pods.

n: This walkthrough uses billable components of Google Cloud. Check the Cloud TPU pricing page (/tpu/docs/pricin te your costs. Be sure to clean up (#clean_up) resources you create when you've finished with them to avoid unnece es.

This section provides information on setting up Cloud Storage storage, VM, and Cloud TPU resources for tutorials.

ant: Set up all resources in the same region/zone to reduce network latency and network costs.

You need a Cloud Storage bucket to store the data you use to train your model and the training results. The `ctpu up` tool used in this tutorial sets up default permissions for the Cloud TPU service account. If you want finer-grain permissions, review the access level permissions (/tpu/docs/storage-buckets).

The bucket location must be in the same region as your virtual machine (VM) and your TPU node. VMs and TPU nodes are located in specific zones (/tpu/docs/types-zones#types), which are subdivisions within a region.

1. Go to the Cloud Storage page on the Cloud Console.

Go to the Cloud Storage page (https://console.cloud.google.com/storage/browser)

2. Create a new bucket, specifying the following options:
    - A unique name of your choosing.
    - Default storage class: `Standard`
    - Location: Specify a bucket location in the same region where you plan to create your TPU node. See TPU types and zones (/tpu/docs/types-zones#types) to learn where various TPU types are available.

This section demonstrates using the Cloud TPU provisioning tool (https://github.com/tensorflow/tpu/tree/master/tools/ctpu) (`ctpu`) for creating and managing Cloud TPU project resources. The resources are comprised of a virtual machine (VM) and a Cloud TPU resource that have the same name. **These resources must reside in the same region/zone as the bucket you just created.**

You can also set up your VM and TPU resources using `gcloud` commands or through the Cloud Console (https://console.cloud.google.com/). For more information, see the creating and deleting TPUs (/tpu/docs/creating-deleting-tpus) page for details.

1. Open a Cloud Shell window.

    Open Cloud Shell (https://console.cloud.google.com/?cloudshell=true)

2. Run `gcloud config set project <Your-Project>` to use the project where you want to create Cloud TPU.
3. Run `ctpu up` specifying the flags shown for either a Cloud TPU device or Pod slice. Refer to CTPU Reference (/tpu/docs/ctpu-reference) for flag options and descriptions.
4. Set up either a Cloud TPU device or a Pod slice:

The `ctpu up` command creates a virtual machine (VM) and Cloud TPU services.

From this point on, a prefix of `(vm)$` means you should run the command on the Compute Engine VM instance.

When the `ctpu up` command has finished executing, verify that your shell prompt is `username@tpuname`, which shows you are logged into your Compute Engine VM.

T2T conveniently packages data generation for many common open-source datasets in its `t2t-datagen` script. The script downloads the data, preprocess it, and makes it ready for training. To do so, it needs at least 200 GB of local disk space.

You can skip this step if you used `ctpu up` to create your Compute Engine VM since it provides 250 GB of disk space for your VM. If you set up your Compute Engine VM using `gcloud` commands or the Cloud Console, and did not specify the VM disk size to be at least 200 GB, follow the instructions below.

- Follow the Compute Engine guide to add a disk (/compute/docs/disks/add-persistent-disk) to your Compute Engine VM.
- Set the disk size to 200 GB (the recommended minimum size).
- Set **When deleting instance** to **Delete disk** to ensure that the disk is removed when you remove the VM.

Make a note of the path to your new disk. For example: `/mnt/disks/mnt-dir`.

On your Compute Engine VM:

1. Create the following environment variables:

   where:

   - `YOUR-BUCKET-NAME` is the name of your Cloud Storage bucket.
   - `DATA_DIR` is a location on Cloud Storage that holds the training and evaluation data.
   - `YOUR-TMP_DIRECTORY` is a location to use to store temporary data. If you added a disk to your Compute Engine VM, this will be a location on the added disk (for example, `/mnt/disks/mnt-dir/t2t_tmp`. Otherwise, it will be a temporary directory on your VM (for example, `/tmp/t2t_tmp`).

2. If you added a new disk to your Compute Engine VM, create a temporary directory on the added disk.

★ **Note:** You do not need to create a temporary directory if you are using a local directory on your Compute Engine VM.

3. Add the path to `tensor2tensor` scripts used to process the model data:

4. Use the `t2t-datagen` (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/bin/t2t-datagen) script to generate the training and evaluation data on the Cloud Storage bucket, so that the Cloud TPU can access the data:

Downloading, preprocessing, and uploading to Cloud Storage takes approximately 2 hours. You can view the data on Cloud Storage:

1. Navigate to the Google Cloud Console.
2. Select **Storage** from the left-hand menu.
3. Click the name of the bucket you created for this tutorial.

You should see sharded files named `translate_ende_wmt32k_packed-train` and `translate_ende_wmt32k_packed-dev`.

You can use the `transformer` model for language modeling. To generate the training data and specify the output file, run the following commands:

This download, preprocessing, and upload to Cloud Storage takes approximately two hours.

To train and evaluate the model on a single Cloud TPU device or pod, run the following command:

Run the following commands on your Compute Engine VM:

1. Set up an environment variable for the training directory, which must be a Cloud Storage location:

2. Run `t2t-trainer` (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/bin/t2t-trainer) to train and evaluate the model:

The above command runs 10 training steps, then 3 evaluation steps. It runs in approximately 5 minutes on a v3-8 TPU node. You can (and should) increase the number of training steps by adjusting the `--train_steps` flag. Translations usually begin to be reasonable after ~40k steps. The model typically converges to its maximum quality after ~250k steps.

3. View the output in your Cloud Storage bucket by going to the Google Cloud Console and choosing **Storage** from the left-hand menu. Click the name of the bucket that you created for this tutorial. Within the bucket, navigate to the training directory, for example, **/training/transformer_ende_1**, to see the model output.
4. To see training and evaluation metrics, launch TensorBoard (/tpu/docs/tensorboard-setup) and point it at the training directory in Cloud Storage.

You can use the `transformer_encoder` model for sentiment classification. Run the following commands to generate the training data and specify the output file:

Run the following command to train and evaluate the model:

The above command runs 10 training steps, then 3 evaluation steps. It runs in approximately 5 minutes on a v3-8 TPU node. This model achieves approximately 85% accuracy after approximately 2,000 steps.

To avoid incurring charges to your GCP account for the resources used in this topic:

1. Disconnect from the Compute Engine VM:

   Your prompt should now be `user@projectname`, showing you are in the Cloud Shell.

2. In your Cloud Shell, run `ctpu delete` with the –zone flag you used when you set up the Cloud TPU to delete your Compute Engine VM and your Cloud TPU:

★ **Important:** If you set the TPU resources name when you ran `ctpu up`, you must specify that name with the `--name` flag when you run `ctpu delete` in order to shut down your TPU resources.

3. Run `ctpu status` to make sure you have no instances allocated to avoid unnecessary charges for TPU usage. The deletion might take several minutes. A response like the one below indicates there are no more allocated instances:

4. Run `gsutil` as shown, replacing ***bucket-name*** with the name of the Cloud Storage bucket you created for this tutorial:

For free storage limits and other pricing information, see the Cloud Storage pricing guide (/storage/pricing).

- Learn more about **`ctpu`** (https://github.com/tensorflow/tpu/tree/master/tools/ctpu), including how to install it on a local machine.
- Explore more Tensor2Tensor models for TPU (https://github.com/tensorflow/tensor2tensor/blob/master/docs/cloud_tpu.md).
- Experiment with more TPU samples (/tpu/docs/tutorials/).
- Explore the TPU tools in TensorBoard (/tpu/docs/cloud-tpu-tools).