

Alpha

This product or feature is in a pre-release state and might change or have limited support. For more information, see the [product launch stages \(/products/#product-launch-stages\)](/products/#product-launch-stages).

This tutorial shows you how to train the Bidirectional Encoder Representations from Transformers (BERT) model on Cloud TPU.

BERT is a method of pre-training language representations. Pre-training refers to how BERT is first trained on a large source of text, such as Wikipedia. You can then apply the training results to other Natural Language Processing (NLP) tasks, such as [question answering](https://en.wikipedia.org/wiki/Question_answering) (https://en.wikipedia.org/wiki/Question_answering) and [sentiment analysis](https://en.wikipedia.org/wiki/Sentiment_analysis) (https://en.wikipedia.org/wiki/Sentiment_analysis). With BERT and Cloud TPU, you can train a variety of NLP models in about 30 minutes.

For more information about BERT, see the following resources:

- [Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing](https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html) (<https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>)
 - [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](https://arxiv.org/abs/1810.04805) (<https://arxiv.org/abs/1810.04805>)
-
- Create a Cloud Storage bucket to hold your dataset and model output.
 - Clone the BERT repository and other required files.
 - Run the training job.
 - Verify the output results.

This tutorial uses billable components of Google Cloud, including:

- Compute Engine
- Cloud TPU
- Cloud Storage

Use the [pricing calculator \(/products/calculator/\)](/products/calculator/) to generate a cost estimate based on your projected usage. New Google Cloud users might be eligible for a [free trial \(/free/\)](/free/).

This section provides information on setting up Cloud Storage storage and a Compute Engine VM.

Important: Set up your Compute Engine VM, your Cloud TPU node and your Cloud Storage bucket in the same region/zone to reduce network latency and network costs.

1. Open a Cloud Shell window.

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

2. Create a variable for your project's name.

3. Configure `gcloud` command-line tool to use the project where you want to create Cloud TPU.

4. Create a Cloud Storage bucket using the following command:

★ **Note:** In the following command, replace **bucket-name** with the name you want to assign to your bucket.

This Cloud Storage bucket stores the data you use to train your model and the training results.

In order for the Cloud TPU to read and write to the storage bucket, the service account for your project needs read/write or Admin permissions on it. See the section on [storage buckets](https://cloud.google.com/tpu/docs/storage-buckets) (<https://cloud.google.com/tpu/docs/storage-buckets>) for how to view and set those permissions.

5. Launch a Compute Engine VM and Cloud TPU using the `ctpu up` command.

★ **Note:** If you have more than one project, you must specify the project name with the `--project` flag. If the Compute Engine VM name is not specified with the `--name` flag, it defaults to your username.

6. The configuration you specified appears. Enter `y` to approve or `n` to cancel.

7. When the `ctpu up` command has finished executing, verify that your shell prompt has changed from `username@project` to `username@tpuname`. This change shows that you are now logged into your Compute Engine VM.

★ **Note:** If you are not connected to the Compute Engine instance, you can connect by running the following commands, replacing **vm-name** with your VM name. The VM name defaults to your username:

As you continue these instructions, run each command that begins with `(vm)$` in your VM session window.

Note: The Compute Engine VM has model code pre-downloaded from [GitHub](https://github.com/tensorflow/models) (<https://github.com/tensorflow/models>).

1. From your Compute Engine virtual machine (VM), install `requirements.txt`.

2. Optional: download `download_glue_data.py`

This tutorial uses the General Language Understanding Evaluation (GLUE) benchmark to evaluate and analyze the performance of the model. The GLUE data is provided for this tutorial at <gs://cloud-tpu-checkpoints/bert/classification>.

If you want to work with raw glue data and create TFRecords, follow the dataset processing instructions on [GitHub](#)

([https://www.google.com/url?](https://www.google.com/url?sa=D&q=https%3A%2F%2Fgithub.com%2Ftensorflow%2Fmodels%2Ftree%2Fmaster%2Fofficial%2Fnlp%2Fbert%23process-datasets)

<https://www.google.com/url?sa=D&q=https%3A%2F%2Fgithub.com%2Ftensorflow%2Fmodels%2Ftree%2Fmaster%2Fofficial%2Fnlp%2Fbert%23process-datasets>)

.

Next, define several parameter values that are required when you train and evaluate your model:

Parameter	Description
STORAGE_BUCKET	This is the name of the Cloud Storage bucket that you created in the **Before you begin** section.
BERT_BASE_DIR	This is the directory that contains files used for training.
MODEL_DIR	This is the directory that contains the model files. This tutorial uses a folder within the Cloud Storage bucket. You do not have to create this folder beforehand. The script creates the folder if it does not already exist.
GLUE_DIR	This is the directory that contains the GLUE training data.

From your Compute Engine VM, run the following command.

The training takes approximately 1 hour on a v3-8 TPU. When script completes, you should see results similar to the following:

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:

Note: For free storage limits and other pricing information, see the [Cloud Storage pricing guide](#) (/storage/pricing).