Alpha

This 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).

For information about access to this release, see the access request page (http://www.google.com).

This tutorial shows you how to train the ResNet-50 model on a Cloud TPU device with PyTorch. You can apply the same pattern to other TPU-optimised image classification models that use PyTorch and the ImageNet dataset.

The model in this tutorial is based on Deep Residual Learning for Image Recognition (https://arxiv.org/pdf/1512.03385.pdf), which first introduces the residual network (ResNet) architecture. The tutorial uses the 50-layer variant, ResNet-50, and demonstrates training the model using PyTorch/XLA (https://github.com/pytorch/xla).

Warning: This tutorial uses a third-party dataset. Google provides no representation, warranty, or other guarantees about the validity, or any other aspects of this dataset.

- Prepare the dataset.
- Run the training job.
- Verify the output results.

This tutorial uses billable components of Google Cloud, including:

- Compute Engine
- Cloud TPU

Use the pricing 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/).

Before starting this tutorial, 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).

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

Go to the console (https://console.cloud.google.com/compute/instancesAdd) and create a new VM instance, providing a name and zone for the VM.

If you plan on training Resnet50 on real data, choose the machine type (https://cloud.google.com/compute/docs/machine-types) with the highest number of CPUs that you can. Resnet50 is typically highly input-bound so the training can be quite slow unless there are many workers to feed in data and sufficient RAM to maintain a large number of worker threads. For best results, select `n1-highmem-96` machine type.

Note: Check your CPU quota to be sure you have enough CPUs available. To increase your quota, see CPU quota (#cpu_quota).

Select the `Debian GNU/Linux 9 Stretch + PyTorch/XLA` for your Boot Disk. If you plan to download ImageNet, specify a disk size of at least 300GB. If you plan to only use fake data, you can specify the default disk size of 20GB.

This tutorial suggests using both data sets.

1. Go to the console (https://console.cloud.google.com/compute/tpus/add) to create a TPU.
2. Under "Name", specify a name for your TPU Pod.
3. Under "Zone" specify the zone (/tpu/docs/types-zones) to use for your Cloud TPU. Make sure it is in the same zone as your VM.
4. Under "TPU type", select the Cloud TPU type. (/tpu/docs/types-zones). For best results, select the `v3-8` TPU.
5. Under "TPU software version" select the latest stable release (`pytorch-0.5`).
6. Use the default network.
7. Set the IP address range (/tpu/docs/internal-ip-blocks). For example, 10.240.0.0.

We recommend using the fake data set for your initial run instead of the real ImageNet set since `fake_data` is automatically installed in your VM, and requires less time and fewer resources to process.

You can use either conda or Docker for your training. Use conda if you are not familiar with Docker.

- Run the version of the code packaged in a conda environment within the `Debian GNU/Linux 9 Stretch + PyTorch/XLA` disk.

- Run using Docker. It will take a few minutes to pull the latest docker image before you can start training.

Option 1

Option 2

If everything looks OK using the `--fake_data` flag, you can try training on real data, such as ImageNet (http://image-net.org/download).

In general, `test_train_imagenet.py` uses `torchvision.datasets.ImageFolder` so you can use any dataset that is structured properly. See the ImageFolder (https://pytorch.org/docs/stable/torchvision/datasets.html#imagefolder) documentation.

Some suggested modifications for real data, assuming you stored the dataset at **~/imagenet**:

Option 1

Option 2

On the full ImageNet dataset with `n1-highmem-96` VM and `v3-8 TPU`, it typically takes ~20 minutes for the first epoch and ~11 minutes for subsequent epochs. The model should reach approximately 76% top-1 accuracy at 90 epochs.

Plan and request additional resources a few days in advance to ensure that there is enough time to fulfill your request.

1. Go to the **Quotas** page.

   Go to the Quotas page (https://console.cloud.google.com/iam-admin/quotas)

2. From the **Service** menu, select **Cloud TPU API**.
3. Select the region or zones where you want to use the CPUs.
4. From the **Metric** menu, select None and then enter CPUs in the search box.
5. Select **CPUs**.
6. In the list, select **Compute Engine API - CPUs**, then click **Edit Quotas** at the top of the page.
7. Enter the amount of quota you are requesting and a description (required), then click **Done**. A request is sent to your service provider for approval.

To avoid incurring charges to your Google Cloud Platform account for the resources used in this tutorial:

1. Exit from the Compute Engine VM.
2. Delete the VM (https://console.cloud.google.com/compute/instances/).
3. Delete the TPU (https://console.cloud.google.com/compute/tpus/).

Try the PyTorch colabs:

- Training MNIST on TPUs (https://colab.sandbox.google.com/github/pytorch/xla/blob/master/contrib/colab/mnist-training-xrt-1-15.ipynb)
- Training ResNet18 on TPUs with Cifar10 dataset (https://colab.sandbox.google.com/github/pytorch/xla/blob/master/contrib/colab/resnet18-training-xrt-1-15.ipynb)
- Inference with Pretrained ResNet50 Model (https://colab.sandbox.google.com/github/pytorch/xla/blob/master/contrib/colab/resnet50-inference-xrt-1-15.ipynb)