CMLG Course Lab I

Haoteng Yin, Shu Li
CS@Purdue

Outline

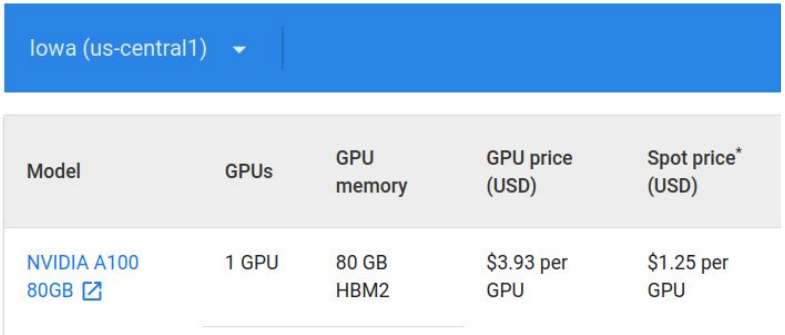
- How to use Google Cloud
- Build a GNN with PyG
- Q & A

How to use Google Cloud for computation?

Get Ready:

- Sign up for Google Cloud Platform
- Google Cloud Platform welcome page
- Creates a project and note the Project ID
- Set up a billing account and Link to your project
- Redeem Google Cloud Education Credit (see email earlier)

Create a VM with attached GPUs



Before you begin

- Read about GPU pricing on Compute Engine to understand the cost to use GPUs on your VMs.
- Read about restrictions for VMs with GPUs.
- Check your GPU quota:
  - New projects have a global GPU quota, which limits the total number of GPUs you can create in any supported zone.

Check your GPU quota

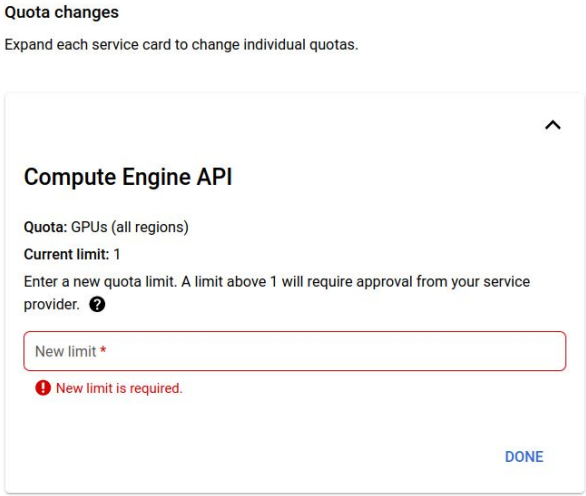
- Use the regions describe command to ensure that you have sufficient GPU quota in the region where you want to create VMs with GPUs.

```
$ gcloud compute regions describe REGION
```

- If you need additional GPU quota, request a quota increase.
  - When you request a GPU quota, you must request a quota for
    - the GPU types that you want to create in each region
    - an additional global quota for the total number of GPUs of all types in all zones.

Request a GPU quota increase

- In the Google Cloud console, Go to Quotas
- Click filter_list **Filter table** and select **Service**.
- Choose **Compute Engine API**.
- Choose **Limit Name: GPU-ALL-REGIONS-per-project**.
- Click the checkbox of the region whose quota you want to change.
- Click create **Edit Quotas**.
- Complete the form.
- Click **Submit Request**. (normally within 5 mins)



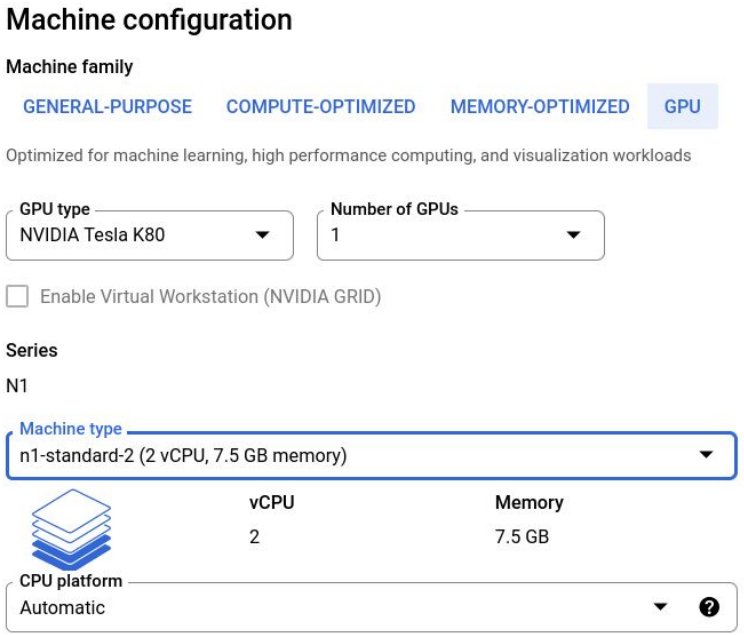
Create a virtual machine with attached GPUs

- In the Google Cloud console, Go to Create an instance
- Specify a **Name** for your VM.
- Select a region and zone that supports the GPU model.
    - To learn about GPU models supported by different regions and zones, see GPU regions and zones availability.

Machine configuration

Complete the following steps:

- In **Machine family**, select **GPU** and the **Series** will be **N1**.
  - Specify the GPU type and number of GPUs.
- In **Machine type**, select a N1 machine type. Alternatively, you can specify custom machine type settings. [standard, high RAM/CPU]
- E.g. Nvidia K80, N1-Standard-2: vCPU #2, Memory 7.5 GB



Choose an operating system image

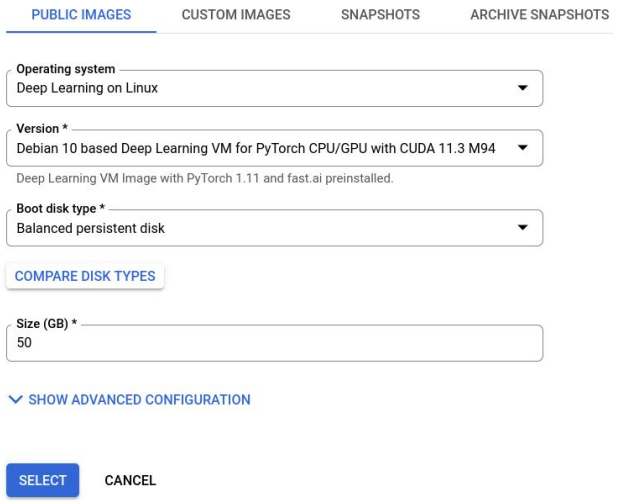
- To select your operating system, in the **Boot disk** section, click **Change**. This opens the **Boot disk configuration** page.
- On the **Boot disk configuration** page, do the following:
  - On the **Public images** tab, choose a supported Compute Engine image or Deep Learning VM image.
  - Specify a boot disk size of at least 40 GB.
  - To confirm your boot disk options, click **Save**.

Switch to images with preinstalled CUDA stack

Use a Deep Learning VM image with GPU drivers pre-installed and include packages, such as TensorFlow / PyTorch.

- OS: Deep Learning on Linux
- Version: M94 with PyTorch 1.11 and fast.ai preinstalled
- Disk Size: > 40 GB

[!] You may need to reinstall Nvidia driver after the first initialization of VM.



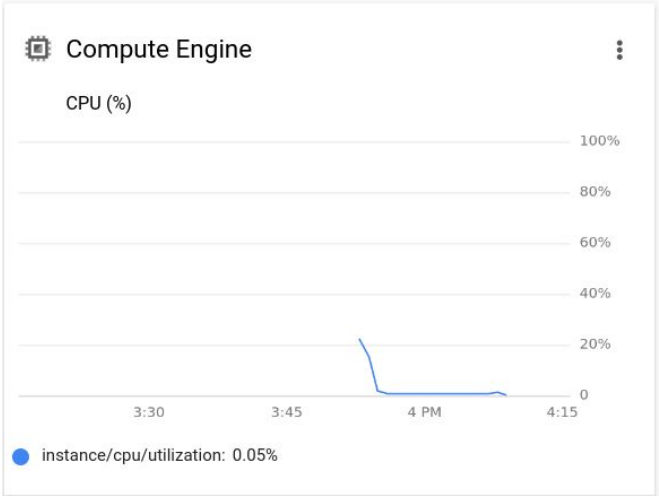
Machine configuration

- Configure other VM settings that you require.
  - Firewall and Network Traffic
  - Change the **Preemptibility / Spot** settings to configure your VM as a preemptible VM.
- To create and start the VM, click **Create**.

Prepare the instance

- Allow a short period of time for the instance to start. After the instance is ready, it's listed on the **VM instances** page with a green status icon.
- Compute Engine grants the user who creates the VM with the roles/compute.instanceAdmin role, and adds that user to the sudo group.

[!] It is highly recommended to keep close eyes for your running instances and monitoring their metrics and maximize their utilization for a lower cost.



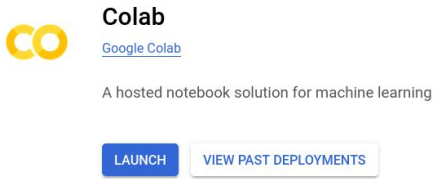
Connect to the VM instance

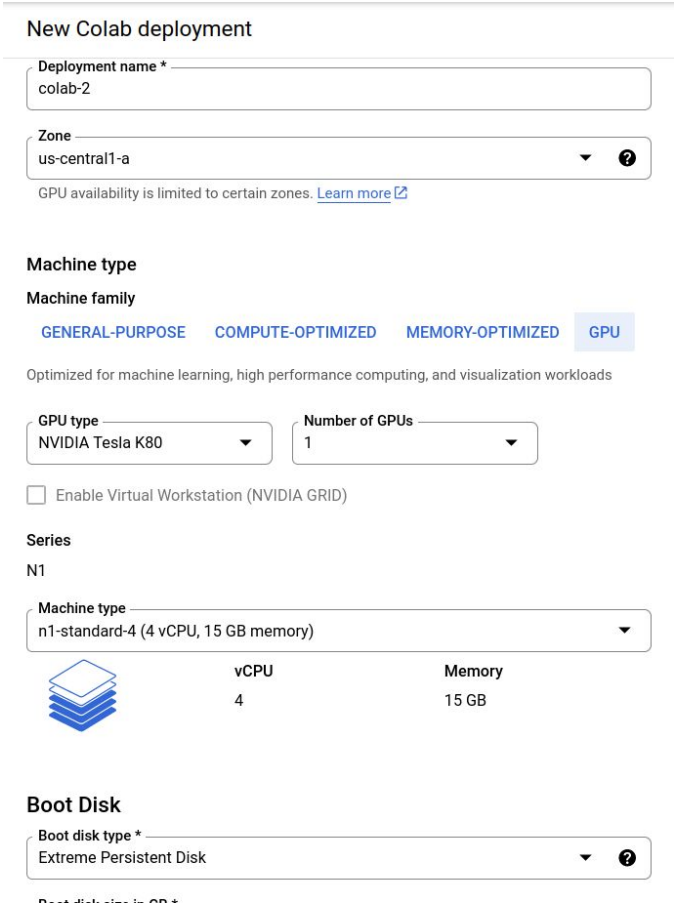
Connect to an instance by using the Google Cloud console and completing the following steps.

- In the Google Cloud console, Go to VM instances
- In the list of virtual machine instances, click **SSH** in the row of the instance that you want to connect to.

How to start a GCE on Colab

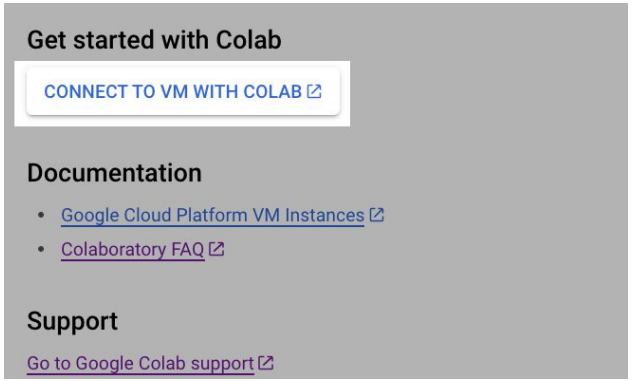
- Colaboratory or ‘Colab’ allows you to write and execute Python in your browser. It allows you to combine executable code and rich text in a single document.
- Visit Colab on GCP Marketplace to launch a preconfigured VM purchase flow.
- Similar **Machine Configuration** but do aware the quota for GPU and Boot Disk. But this can **only** work for Colab.

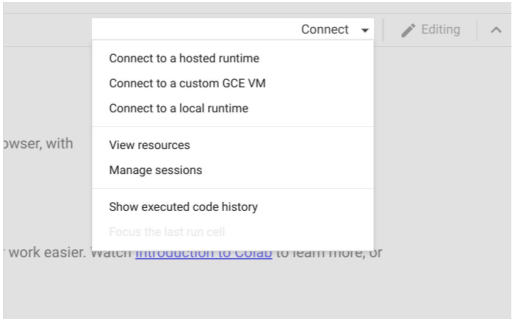


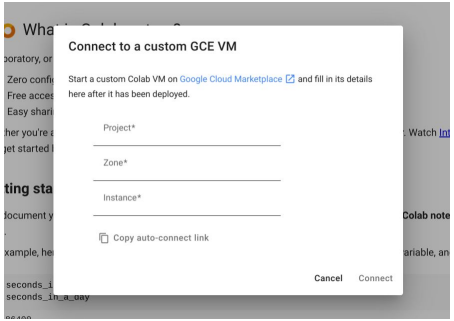


Connect Colab to your GCE VM

- Use the link from within GCP's Deployment Manager.
- The link is present in the details of your Colab deployment.
- If you wish to connect from within Colab, from the Connect arrow within Colab select "Connect to a Custom GCE VM".
- Fill in the resulting dialog with the information from your VM configuration and click Connect.







Build a graph learning pipeline with PyG

PyG is a library built upon PyTorch to easily write and train Graph Neural Networks for a wide range of applications related to structured data.

PyG is both friendly to machine learning researchers and first-time users of machine learning toolkits.

Example: Colab [Link], Runtime Type: GPU

```python
dataset = Planetoid(root='.', name='Cora')

class GCN(torch.nn.Module):
    def __init__(self, in_channels, hidden_channels, out_channels):
        super().__init__()
        self.conv1 = GCNConv(in_channels, hidden_channels)
        self.conv2 = GCNConv(hidden_channels, out_channels)

    def forward(self, x: Tensor, edge_index: Tensor) -> Tensor:
        x = self.conv1(x, edge_index).relu()
        x = self.conv2(x, edge_index)
        return x

model = GCN(dataset.num_features, 16, dataset.num_classes)
```

Q & A