



Introduction to Program 3

ECS 170

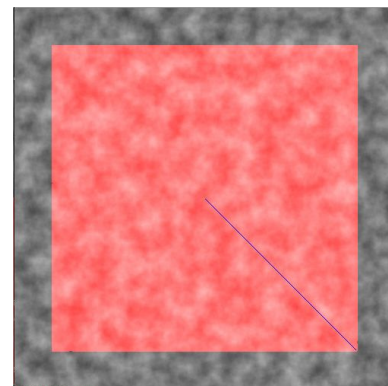
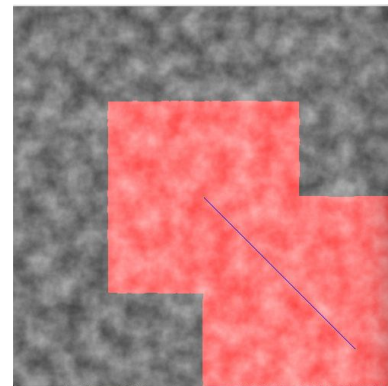
Hannah Brown

Nodes Expanded: 16,662

Time Taken: 12,4542

Bidirectional A*

- Two searches: one starting from the initial state and one from the goal state
- Requires more bookkeeping: an open and closed list for each search, and cost has to be calculated carefully for both $g(n)$ and $h(n)$
 - Priorityqueues used for open lists and hashmaps used for closed lists to minimize bookkeeping cost
- $O(b^{d/2})$ instead of $O(b^d)$
- My heuristic: $h(k) = (k * \text{current height}) / 256$,
 - k = Chebyshev distance from current position to goal
- Admissible but not consistent, so there were some node re-expansions



Bidirectional Dijkstra vs.
Normal Dijkstra

When to Stop Looking

- When the two frontiers intersect, the shortest path is guaranteed to have been found
- The position where the two frontiers intersect is *not guaranteed* to be on that shortest path!
- Keep track of the min cost path seen so far and the nodes in that path
- Terminate when: $\min_cost \leq \max(\min(f_start(x)), \min(f_goal(x)))$
 - x is a node in both frontiers

Termination condition source: <https://arxiv.org/pdf/cs/9712102.pdf>



Bidirectional A* on Mt. St. Helens

Matthew Marlow

Nodes Expanded: 455

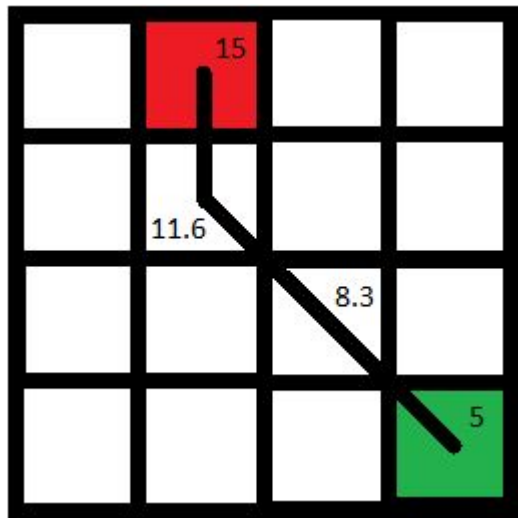
Time Taken: 78,938

Heuristic Function

```
// return 0 if we're comparing the goal state to the goal state
if (pt1.x == pt2.x && pt1.y == pt2.y)
    return 0;

// otherwise calculate the heuristic
// h = avg step cost * min number of steps to goal.
double startHeight = map.getTile(pt1);
double endHeight = map.getTile(pt2);
double delta = endHeight - startHeight;
int chebDist = Math.max(Math.abs(pt2.x - pt1.x), Math.abs(pt2.y - pt1.y));
double avgStepHeight = delta / chebDist;
double h = 0.0;
for (int x = 0; x < chebDist; x++) {
    h += startHeight / (startHeight + avgStepHeight + 1);
    startHeight += avgStepHeight;
}

return h * 0.95;
```



Increasing Efficiency With Bi-Directional Graph Search

Fringe - `TreeMap<Double, LinkedList<Point>>` - automatically inserts in key-order, one for each direction

Closed List - `Boolean[][]` - constant lookup/write time, one for each direction. Stops searching when the first tile in one direction's fringe is in the closed list of the other direction.

G Value Table - `Double[][]` - Like with the closed list, using a 2d array is more space complex, but less time complex. Only 1 required for Bi-Directional search since the two directions will never overlap

Introduction to Google Cloud



0. Redeem Credits

- Link in announcements
- Credits are emailed and get redeemed to **default** Google account
- **Only redeem credits once, there is only enough for 1 coupon per student**
- Worst case there is free trial credits

Here is the URL you will need to access in order to request a Google Cloud Platform coupon. You will be asked to provide your school email address and name. An email will be sent to you to confirm these details before a coupon is sent to you.

[Student Coupon Retrieval Link](#) 

- You will be asked for a name and email address, which needs to match the domain. A confirmation email will be sent to you with a coupon code.
- You can request a coupon from the URL and redeem it until: 5/5/2020
- Coupon valid through: 1/5/2021
- You can only request ONE code per unique email address.

Please contact me if you have any questions or issues.



1. Create a Project



Google Cloud Platform



ucdavis.edu



Dashboard




Page not viewable for organizations. To view this page, select a project.

[SELECT PROJECT](#)

[CREATE PROJECT](#)

2. Project Settings

- Set your new project to match the right
- Make sure to set billing account to ECS170
 - Go through credits redemption first

 Google Cloud Platform

New Project

Project name *

Pong Deep Learner

Project ID: pong-deep-learner. It cannot be changed later. [EDIT](#)

Billing account *

ECS170


Any charges for this project will be billed to the account you select here.

Organization

ucdavis.edu

This project will be attached to ucdavis.edu.

Location *

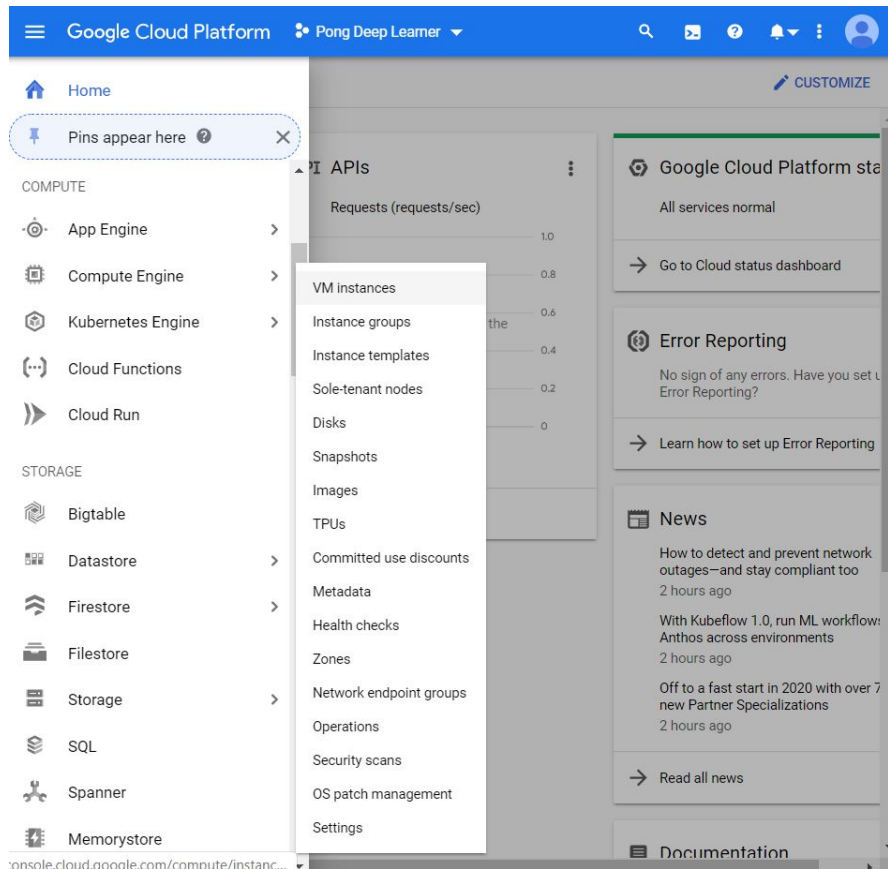
 ucdavis.edu [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

3. Make a VM

- Make a virtual machine
- Hamburger icon
 - Compute Engine
 - VM Instances





4. Wait for Compute Engine to Initialize

Compute Engine is getting ready. This may take a minute or more. [Compute Engine documentation](#)

Click “Create”
when ready

Compute Engine VM instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a migration service, or try the quickstart to build a sample app.

Create

or

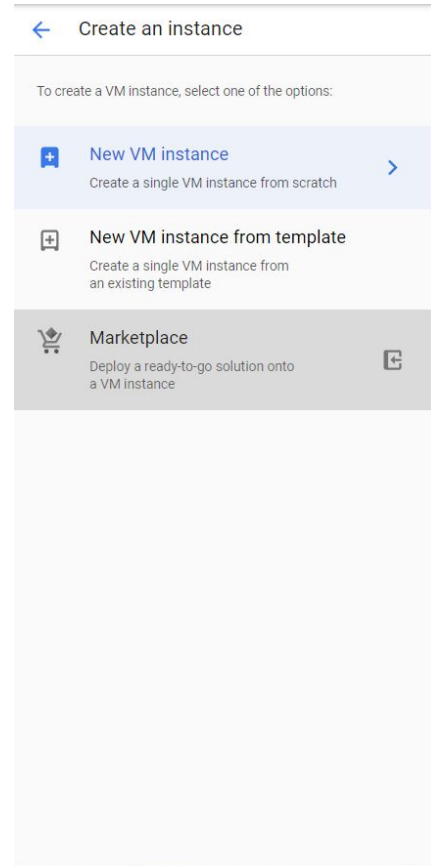
Import

or

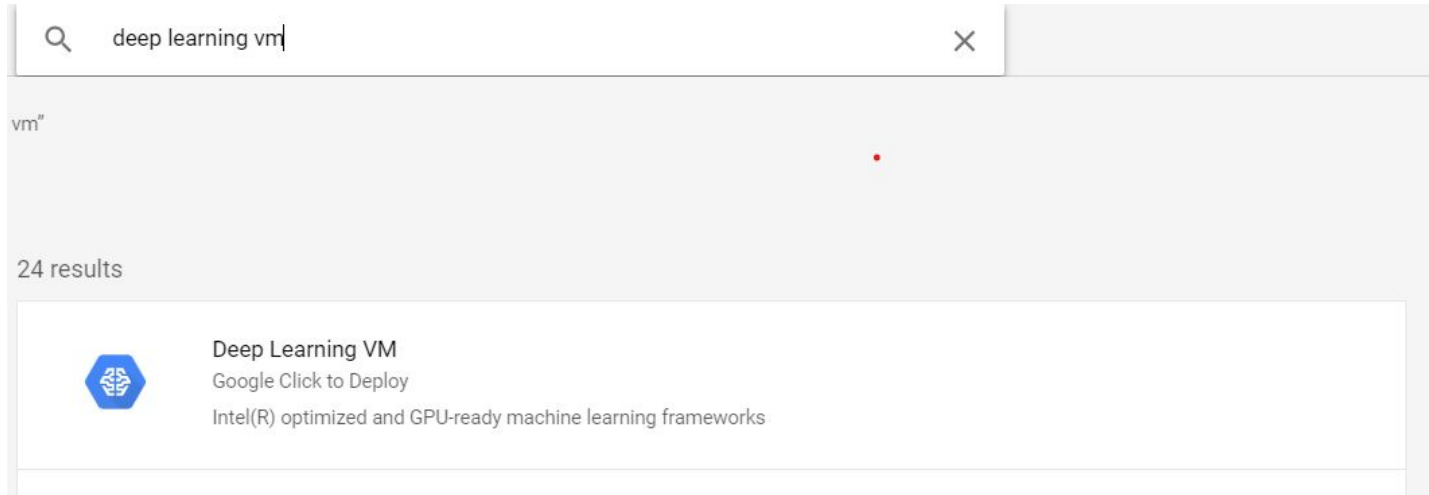
Take the quickstart

5. Making VM

- Click “Marketplace” to get a VM from the marketplace



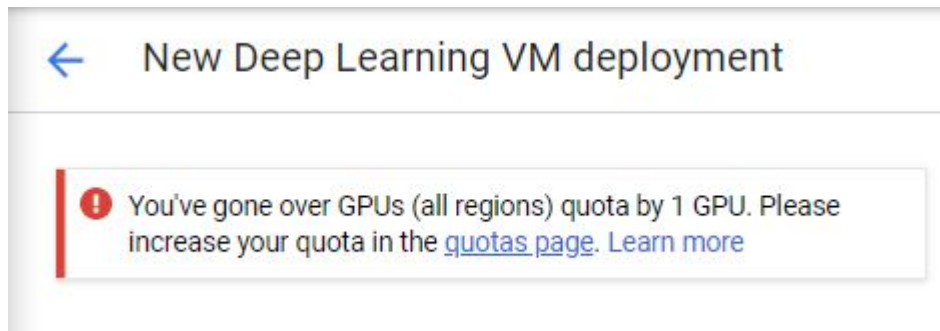
6. Use “Deep Learning VM” from Google





7. GPU Quota

- If you encounter the warning on the right click on “quotas page”
- Neural networks train much faster on GPU’s



8. GPU Quota Request

The screenshot displays the Google Cloud Quotas console. On the left, a table lists available quotas. The 'Compute Engine API' quota for 'GPUs (all regions)' is selected. On the right, a modal window titled 'Compute Engine API' shows the details for the selected quota. The 'New quota limit' is set to 1, and the 'Request description' is 'I am a student'.

Quota type	Service	Metric	Location
All quotas	All services	GPUs (all regions)	All locations

Service	Location	Current Usage	7 Day Peak Usage	Limit	
Compute Engine API	Global			0	View hierarchy

1 quota selected

Compute Engine API

Quota: GPUs (all regions)

New quota limit
Enter a new quota limit. Your request will be sent to your service provider for approval.

1

Request description
Required ⓘ

I am a student

Done Cancel

Submit request Back

- Request an increase in GPUs (all regions) to 1 GPU
- This takes roughly 10 minutes for Google to accept (may take longer)

9. VM Settings

- Make your VM match the right after GPU quota increased
- Important
 - GPUs: 1
 - Framework: Tensorflow 2.1
- (This is the page from step 7)

← New Deep Learning VM deployment

Deployment name

tensorflow-1

Zone

GPU availability is limited to certain zones. [Learn more](#)

us-west1-b

Machine type

2 vCPUs

13 GB memory

[Customize](#)

GPUs

The number of GPU dies is linked to the number of CPU cores and memory selected for this instance. For the current configuration, you can select no fewer than 1 GPU die of this type. [Learn more](#)

Number of GPUs

1

GPU type

NVIDIA Tesla K80

1

Machines with GPUs can't migrate on host maintenance

Framework

Choose the primary machine learning framework you will be using. If the library you would like to use is not listed, choose the base image, which provides core packages.

TensorFlow Enterprise 2.1 (CUDA 10.1)

GPU

☒ Install NVIDIA GPU driver automatically on first startup?

I want to use NVIDIA GPUs with this image. Please fetch NVIDIA GPU drivers from a third-party location and install them on my behalf (requires internet access on the VM).

[Access to the Jupyter Lab](#)

Deep Learning VM overview

Solution provided by Google Click to Deploy

\$295.20 per month estimated

Effective hourly rate \$0.404 (730 hours per month)

[Details](#)

Software

Operating System

Debian (9)

Documentation

[Official Documentation](#)

[StackOverflow: Deep Learning VM](#)

[Google Group: Deep Learning VM](#)

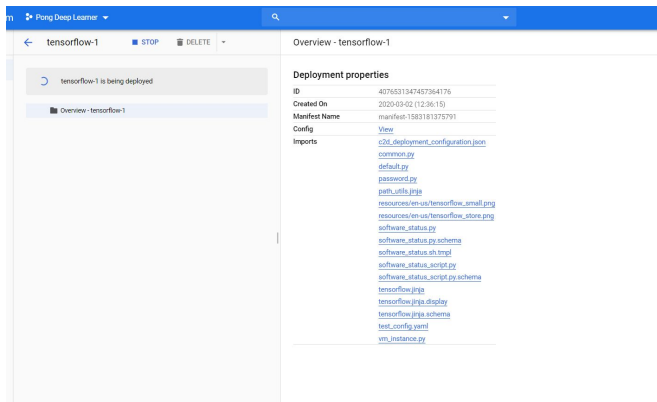
Terms of Service

By using this software or accessing the service you are agreeing to comply with the [GCP Marketplace terms of service](#) and the terms of applicable open source software licenses bundled with the software or service. Please review these terms and licenses carefully for details about any obligations you may have related to the software or service. To the limited extent an open source software license related to the software or service expressly supersedes the GCP Marketplace Terms of Service, that open source software license governs your use of that software or service.

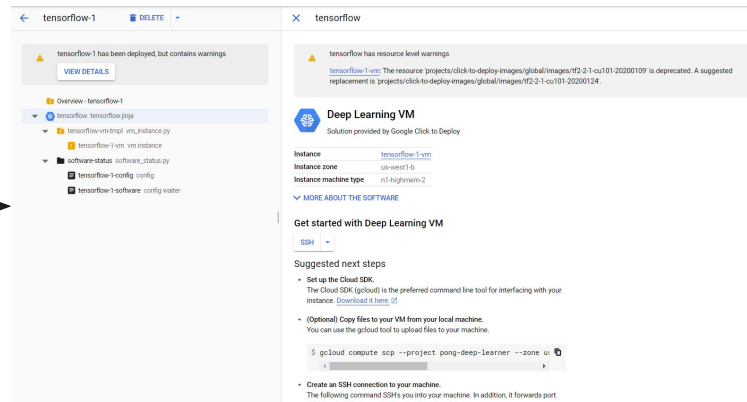
By using this product, you understand that certain account and usage information may be shared with Google Click to Deploy for the purposes of sales attribution, performance analysis, and support.

Google is providing this software or service "as-is" and will not perform any ongoing maintenance. Ongoing upgrades and maintenance are your responsibility.

10. Await Deployment

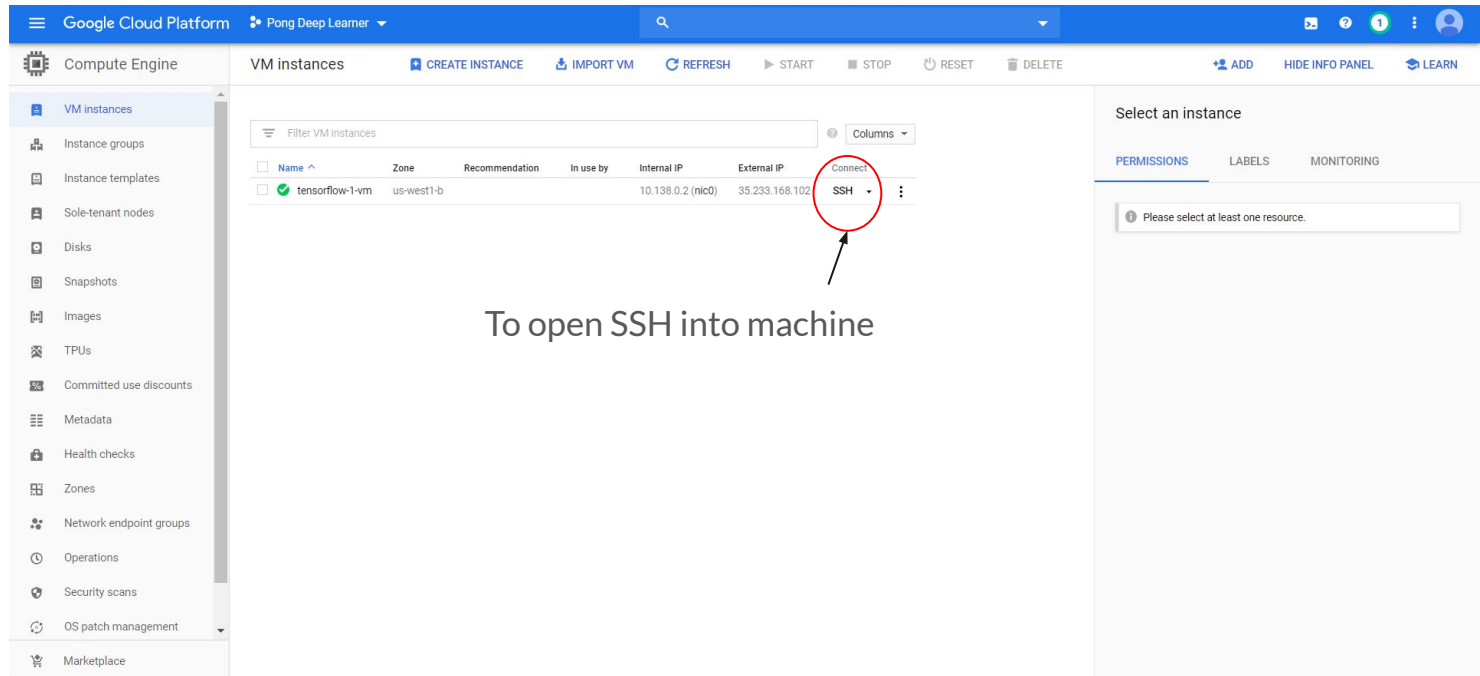


The screenshot shows the Google Cloud Platform console interface. At the top, there's a blue header with 'Pong Deep Learner' and a search bar. Below the header, the left sidebar shows a navigation menu with 'tensorflow-1' selected. The main content area is titled 'Overview - tensorflow-1'. It displays deployment properties for a resource named 'tensorflow-1'. The properties include ID, Created On, Manifest Name, Config, and Imports. The Imports section lists various files and scripts used in the deployment, such as 'click-deployment-configuration.json', 'common.py', 'default.py', 'password.py', 'paths.py', 'resources/en-us/tensorflow_email.png', 'resources/en-us/tensorflow_store.png', 'software_status.py', 'software_status.py.schema', 'software_status.sh.test', 'software_status_script.py', 'software_status_script.py.schema', 'tensorflow.png', 'tensorflow.png.display', 'tensorflow.png.schema', 'test_config.yaml', and 'vm_instance.py'.



The screenshot shows the Google Cloud Platform console interface with the deployment details for 'tensorflow-1'. The left sidebar shows a navigation menu with 'tensorflow-1' selected. The main content area is titled 'Overview - tensorflow-1'. It displays a tree view of the deployment structure, including 'tensorflow-1-vm', 'tensorflow-1-vm.config', and 'tensorflow-1-vm.config.water'. A warning message at the top states: 'tensorflow-1 has been deployed, but contains warnings'. Below the tree view, there's a section titled 'Deep Learning VM' with a solution provided by Google Click to Deploy. It shows the instance name 'tensorflow-1-vm', the instance zone 'us-west1-b', and the instance machine type 'n1-highmem-2'. There's also a section titled 'Get started with Deep Learning VM' with a link to 'SSH'. Below this, there's a section titled 'Suggested next steps' with a list of steps: 'Set up the Cloud SDK', '(Optional) Copy files to your VM from your local machine', and 'Create an SSH connection to your machine'. The 'SSH' section includes a terminal snippet showing the command to connect to the VM: 'gcloud compute scp --project pong-deep-learner --zone us-west1-b tensorflow-1-vm:/root/.ssh/authorized_keys .'

11. SSH into VM



The screenshot shows the Google Cloud Platform console interface. The left sidebar contains a navigation menu with options like 'Instance groups', 'Instance templates', 'Sole-tenant nodes', 'Disks', 'Snapshots', 'Images', 'TPUs', 'Committed use discounts', 'Metadata', 'Health checks', 'Zones', 'Network endpoint groups', 'Operations', 'Security scans', 'OS patch management', and 'Marketplace'. The main content area is titled 'VM instances' and includes a table of instances. The table has columns for Name, Zone, Recommendation, In use by, Internal IP, External IP, and Connect. The first instance listed is 'tensorflow-1-vm' in the 'us-west1-b' zone, with internal IP '10.138.0.2 (nic0)' and external IP '35.233.168.102'. The 'Connect' dropdown menu is open, showing 'SSH' as an option, which is circled in red. An arrow points to the 'SSH' option. The text 'To open SSH into machine' is overlaid on the image.

Google Cloud Platform Pong Deep Learner

Compute Engine VM instances CREATE INSTANCE IMPORT VM REFRESH START STOP RESET DELETE ADD HIDE INFO PANEL LEARN

Filter VM instances Columns

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
tensorflow-1-vm	us-west1-b			10.138.0.2 (nic0)	35.233.168.102	SSH

Select an instance

PERMISSIONS LABELS MONITORING

Please select at least one resource.

To open SSH into machine

11. SSH into VM

Google Cloud Platform

Compute Engine

VM instances

CREATE INSTANCE IMPORT VM REFRESH START STOP RESET DELETE

ADD HIDE INFO PANEL LEARN

Filter VM instances Columns

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
tensorflow-1-vm	us-west1-b			10.138.0.2 (nic0)	35.233.168.102	SSH

Select an instance

PERMISSIONS LABELS MONITORING

Please select at least one resource.

To stop machine (and stop using credits)



Loading the Homework

- For ease of use, the homework is duplicated on GitHub:
- <https://github.com/kurt-schneider/ECS-170-Program-3-Starter-Code>
- `git clone 'https://github.com/kurt-schneider/ECS-170-Program-3-Starter-Code'`
- Make sure to do the additional installs listed in the pdf
- The code will not run until you have implemented all the functions required
- You may have to take additional steps to use the “-g” option for `test_dqn_pong.py`. The gui is not required



Miscellanea

- Use GitHub
- Watch your model train for a bit first to make sure its improving
- Train model using 'nohup python3 run_dqn_pong.py'
 - Do **not** stop the machine if you want nohup to keep running
- 'ALE_del' error/warning on termination can be ignored
 - Something in the library seems to have trouble deleting itself

Questions

PyTorch Crash Course



CUDA

- CUDA is how pytorch interacts with the GPU
- GPUs are (generally) better for neural network training



CUDA

- CUDA is how pytorch interacts with the GPU
- GPUs are (generally) better for neural network training
 - **Why?**



CUDA

- CUDA is how pytorch interacts with the GPU
- GPUs are (generally) better for neural network training
 - Why?
 - GPUs optimized for matrix calculations, parallel processing, high data bandwidth (exactly what we need for neural nets)
- `USE_CUDA = torch.cuda.is_available()`



Tensors

- Similar to (numpy) arrays
- Hold pieces of data (e.g. images, network outputs)
- `action = torch.LongTensor(action)`
- `action.size()`
- `action.view(-1, 10)`
- `action.detach().cpu().numpy()`
- `torch.Tensor()` vs `torch.cuda.Tensor()` vs `torch.Tensor().cuda()` ?



Neural Networks

- Models (neural networks) can also be sent to the GPU using `.cuda()` !

```
class Neural_Network(nn.Module):  
    def __init__(self, ):  
        super(Neural_Network, self).__init__()  
        # parameters  
        # TODO: parameters can be parameterized instead of declaring them here  
        self.inputSize = 2  
        self.outputSize = 1  
        self.hiddenSize = 3  
  
        # weights  
        self.W1 = torch.randn(self.inputSize, self.hiddenSize) # 2 X 3 tensor  
        self.W2 = torch.randn(self.hiddenSize, self.outputSize) # 3 X 1 tensor  
  
    def forward(self, X):  
        self.z = torch.matmul(X, self.W1)  
        self.z2 = self.sigmoid(self.z) # activation function  
        self.z3 = torch.matmul(self.z2, self.W2)  
        o = self.sigmoid(self.z3) # final activation function  
        return o
```



Training Neural Networks

```
loss = compute_td_loss(model, target_model, batch_size,  
                        gamma, replay_buffer) # gets loss  
optimizer.zero_grad() # resets the optimizer gradient  
loss.backward() # backpropagates the loss through the net (i.e. update)  
optimizer.step() # updates optimizer parameters
```

Questions

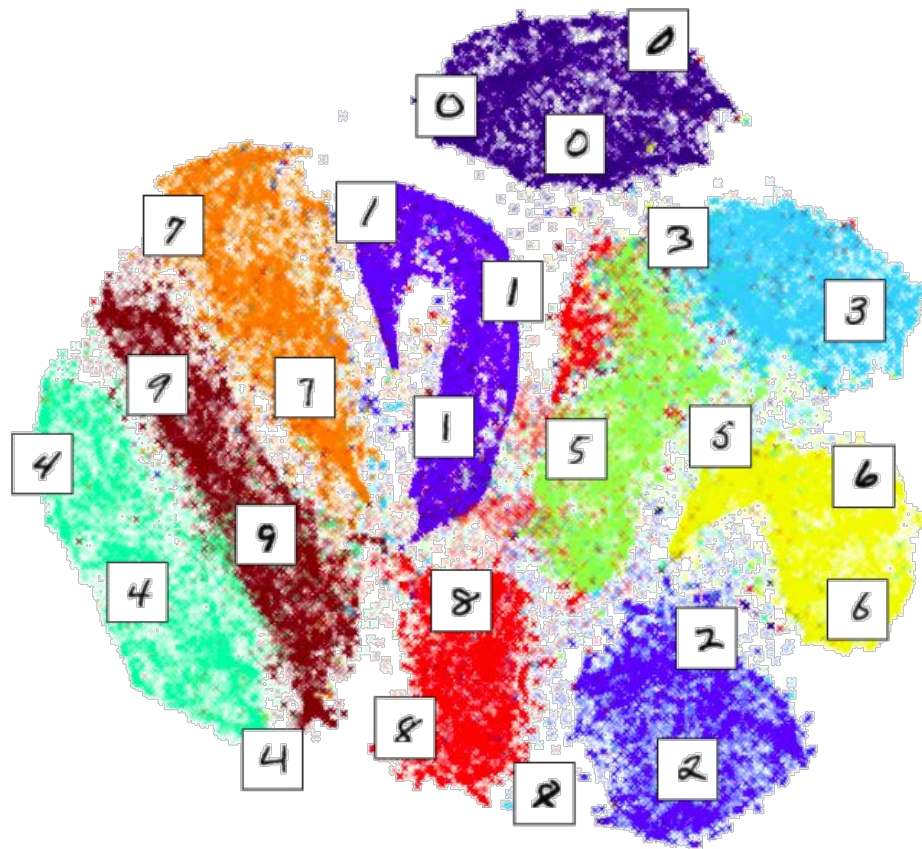


Important Notes

- Training a model takes a while and this is **unavoidable**
 - **Start early!**
 - No trained model => no good grade 😞
- Watch results for a few ten thousand frames
 - No decrease => you probably have a mistake
- Reward **may not** strictly increase, loss **may not** strictly decrease
 - Moving target of minimizing model loss + maximizing reward
 - Target network helps stabilize this
- Try adjusting the parameters
 - E.g. learning rate, epsilon, gamma, etc

Bonus Points

- XAI is an active research area
- Explaining “black box” models is difficult
- Try to use your human knowledge of Pong and observation of the model to explain what it may have learned
 - Give evidence for it with the learned embedding



TSNE Visualization for MNIST Embedding

Good Luck!