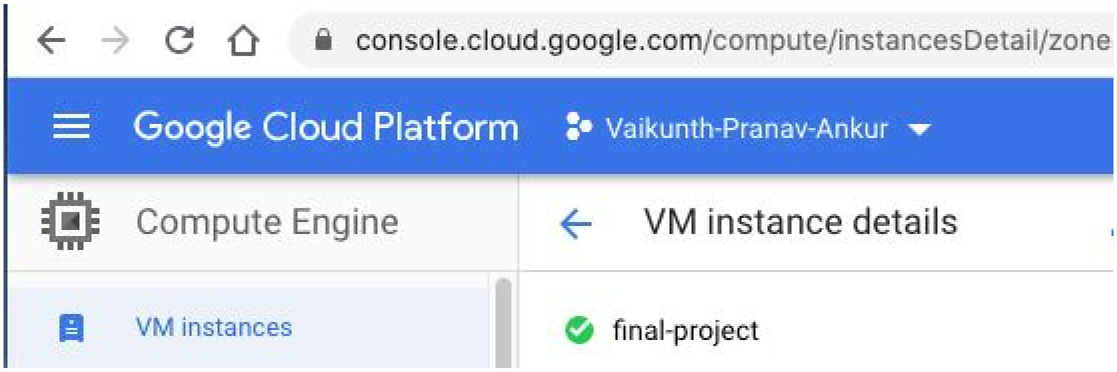
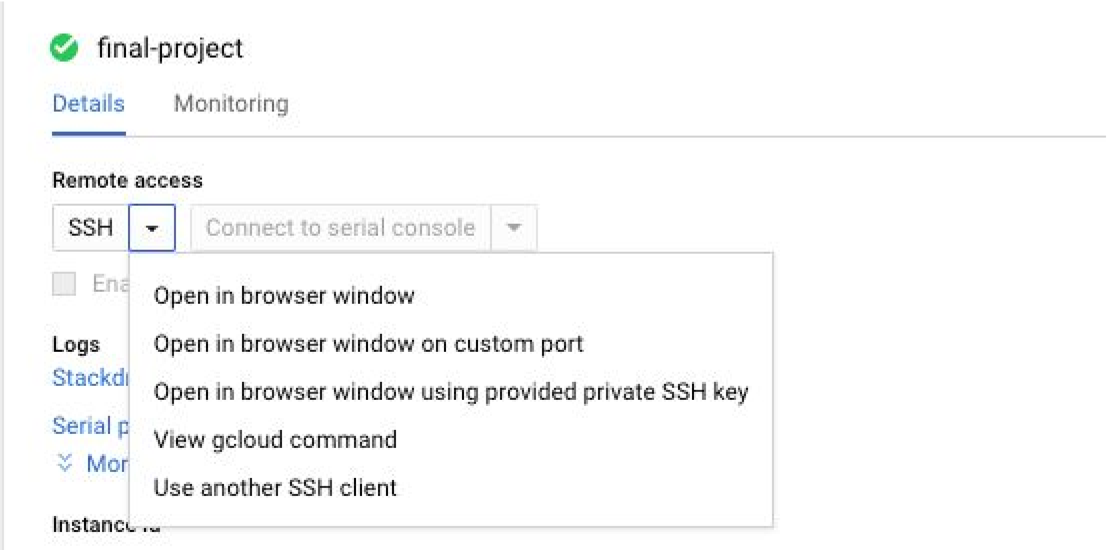
Group-4 Reproducibility Instructions

VM Setup:

- ➔ Go to https://console.cloud.google.com/, and make sure you're signed in with the same gmail account that you provided on the following piazza post.
- ➔ Next, from the project dropdown, select our GCP Project, titled "Vaikunth-Pranav-Ankur".

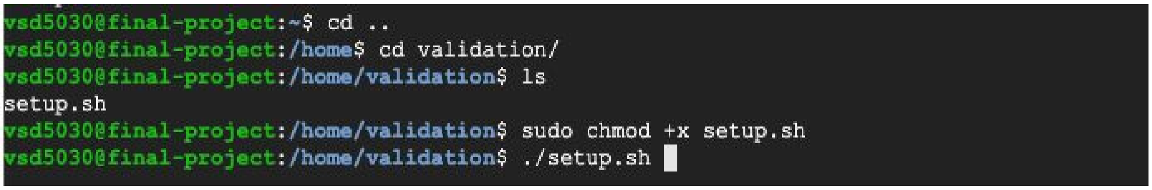


- ➔ Click on the "final-project" VM instance and start running it (if it's off)
- ➔ Click the drop-down arrow next to SSH and select the Open in browser window, as shown in the below screenshot

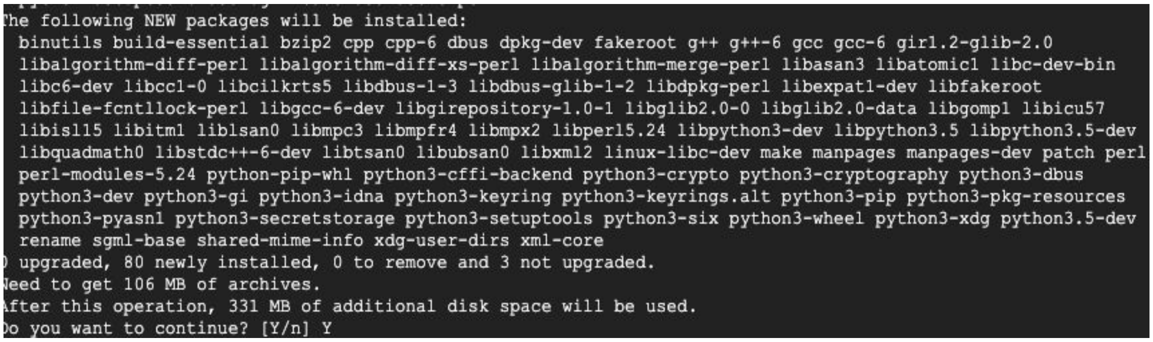


- ➔ On the terminal, you will find that your current home directory would most probably look something like this: /home/your_username. You will need to use “cd ..” to navigate up one directory level and subsequently navigate to the directory called “validation” using the following command: “cd validation”.
- ➔ Once you’re inside the “validation” directory, you can do “ls” to list the files. First, you will run the “setup.sh” file to install all the libraries and dependencies required to run our code. Please follow the commands listed below :

```
sudo chmod +x setup.sh
./setup.sh
```



```
vsd5030@final-project:~$ cd ..
vsd5030@final-project:/home$ cd validation/
vsd5030@final-project:/home/validation$ ls
setup.sh
vsd5030@final-project:/home/validation$ sudo chmod +x setup.sh
vsd5030@final-project:/home/validation$ ./setup.sh
```



```
The following NEW packages will be installed:
  binutils build-essential bzip2 cpp cpp-6 dbus dpkg-dev fakeroot g++ g++-6 gcc gcc-6 gir1.2-glib-2.0
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan3 libatomic1 libc-dev-bin
  libc6-dev libcc1-0 libcilkrts5 libdbus-1-3 libdbus-glib-1-2 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-6-dev libgirepository-1.0-1 libglib2.0-0 libglib2.0-data libgomp1 libicu57
  libisl15 libitm1 liblsan0 libmpc3 libmpfr4 libmpx2 libperl5.24 libpython3-dev libpython3.5 libpython3.5-dev
  libquadmath0 libstdc++-6-dev libtsan0 libubsan0 libxml2 linux-libc-dev make manpages manpages-dev patch perl
  perl-modules-5.24 python-pip-whl python3-cffi-backend python3-crypto python3-cryptography python3-dbus
  python3-dev python3-gi python3-idna python3-keyring python3-keyrings.alt python3-pip python3-pkg-resources
  python3-pyasn1 python3-secretstorage python3-setuptools python3-six python3-wheel python3-xdg python3.5-dev
  rename sgml-base shared-mime-info xdg-user-dirs xml-core
0 upgraded, 80 newly installed, 0 to remove and 3 not upgraded.
Need to get 106 MB of archives.
After this operation, 331 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Please type “Y” when the above prompt appears.

If the setup script fails for some reason, then please run the following commands to install the required dependencies manually:

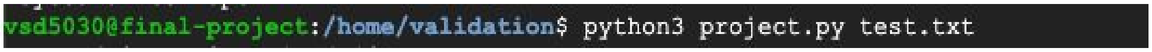
```
sudo apt-get update
sudo apt-get install vim
sudo apt-get install python3-pip
pip3 install networkx
pip3 install numpy
pip3 install matplotlib
pip3 install func_timeout
pip3 install timeout_decorator
```

- Next, please run the following command just to ensure that you’ve the desired write permissions on the current working directory:

```
sudo chmod -R 777 /home/validation/
```

- Once the required setup is completed successfully, please run the following command to run our main script:

```
python3 project.py test.txt
```



```
vsd5030@final-project:/home/validation$ python3 project.py test.txt
```

Please Note*: Our algorithm involves significant randomization and the networkx functions we have used can cause very large time -delays, hence if our algorithm stalls during computation, we request you to wait for a time frame of 10-15 minutes, and then restart computation. In the worst case scenario of randomization, this step may have to be repeated multiple times till the final iteration is reached. (to be precise, if the process is taking longer than the aforementioned time frame to complete, it’s better to kill the process and start again.)

Results and Validation:

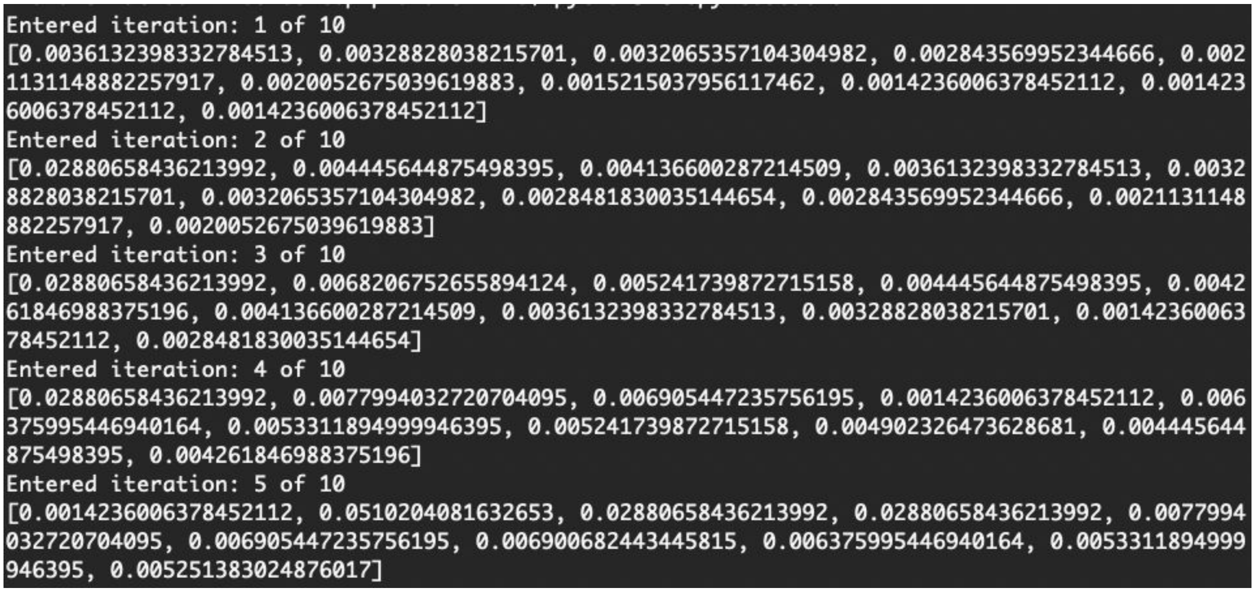
- Upon running our program, the output will be displayed within the terminal itself, and the plot will be updated in the same directory (Plot.png). You can either securely copy (by using the scp command line utility) or download that plot image directly on to your local machine.

- Each iteration of the algorithm provides the current fitness value of the10 networks in the gene population. We mainly run10 such iterations, but the number of iterations can be changed at the user’s discretion.

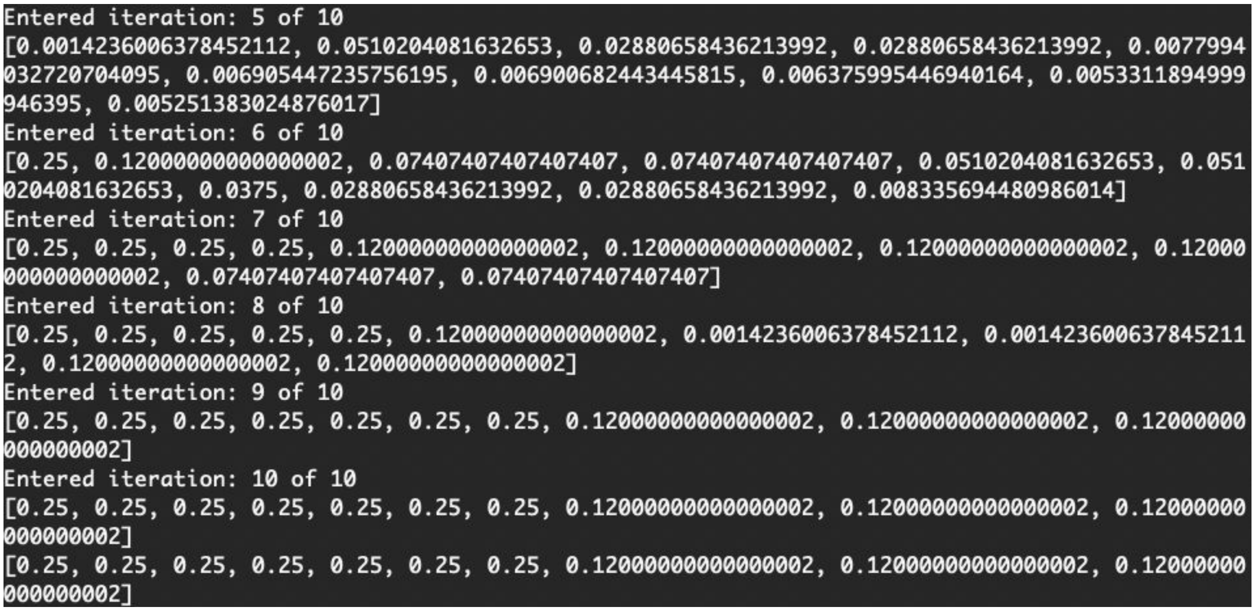
- In most cases, at every iteration of the Genetic algorithm, you will observe improvements in the fitness value of each of the 10 genes(networks), compared to the previous iteration.You can compare the previous value with the current value in each stage.

- The fitness value goes on increasing, generating better networks, until the point where the algorithm converges to a constant optimal value.

- ➔ The output and value updates in each stage of the iteration will be similar to the following output:

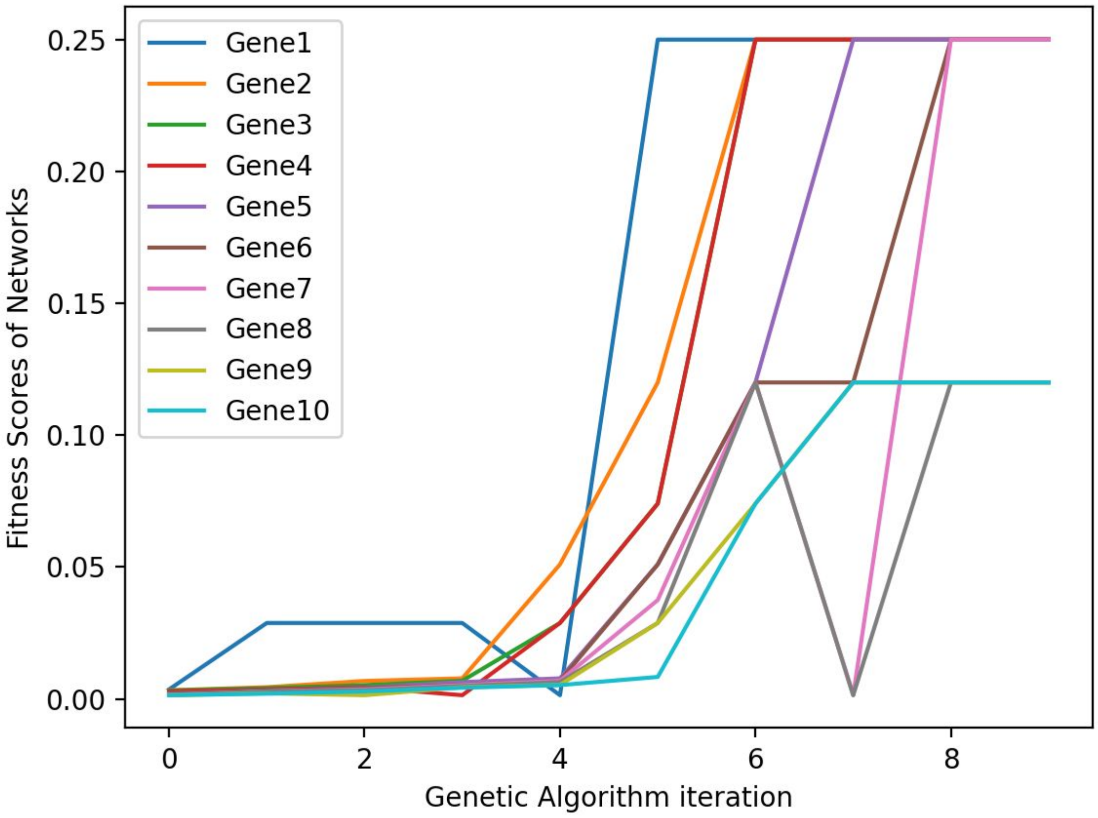


```
Entered iteration: 1 of 10
[0.0036132398332784513, 0.0032882803821570l, 0.0032065357104304982, 0.002843569952344666, 0.002
1131148882257917, 0.0020052675039619883, 0.0015215037956117462, 0.0014236006378452112, 0.001423
6006378452112, 0.0014236006378452112]
Entered iteration: 2 of 10
[0.02880658436213992, 0.004445644875498395, 0.004136600287214509, 0.0036132398332784513, 0.0032
8828038215701, 0.0032065357104304982, 0.0028481830035144654, 0.002843569952344666, 0.0021131148
882257917, 0.0020052675039619883]
Entered iteration: 3 of 10
[0.02880658436213992, 0.0068206752655894124, 0.005241739872715158, 0.004445644875498395, 0.0042
61846988375196, 0.004136600287214509, 0.0036132398332784513, 0.00328828038215701, 0.00142360063
78452112, 0.0028481830035144654]
Entered iteration: 4 of 10
[0.02880658436213992, 0.0077994032720704095, 0.006905447235756195, 0.0014236006378452112, 0.006
375995446940164, 0.0053311894999946395, 0.005241739872715158, 0.004902326473628681, 0.004445644
875498395, 0.004261846988375196]
Entered iteration: 5 of 10
[0.0014236006378452112, 0.0510204081632653, 0.02880658436213992, 0.02880658436213992, 0.0077994
032720704095, 0.006905447235756195, 0.006900682443445815, 0.006375995446940164, 0.0053311894999
946395, 0.005251383024876017]
```



```
Entered iteration: 5 of 10
[0.0014236006378452112, 0.0510204081632653, 0.02880658436213992, 0.02880658436213992, 0.0077994
032720704095, 0.006905447235756195, 0.006900682443445815, 0.006375995446940164, 0.0053311894999
946395, 0.005251383024876017]
Entered iteration: 6 of 10
[0.25, 0.12000000000000002, 0.07407407407407407, 0.07407407407407407, 0.0510204081632653, 0.051
0204081632653, 0.0375, 0.02880658436213992, 0.02880658436213992, 0.008335694480986014]
Entered iteration: 7 of 10
[0.25, 0.25, 0.25, 0.25, 0.12000000000000002, 0.12000000000000002, 0.12000000000000002, 0.12000
000000000002, 0.07407407407407407, 0.07407407407407407]
Entered iteration: 8 of 10
[0.25, 0.25, 0.25, 0.25, 0.25, 0.12000000000000002, 0.0014236006378452112, 0.001423600637845211
2, 0.12000000000000002, 0.12000000000000002]
Entered iteration: 9 of 10
[0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.12000000000000002, 0.12000000000000002, 0.12000000
000000002]
Entered iteration: 10 of 10
[0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.12000000000000002, 0.12000000000000002, 0.12000000
000000002]
[0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.12000000000000002, 0.12000000000000002, 0.12000000
000000002]
```

➔ Each individual gene will converge to an optimal value which is visible in the graph below.



Disclaimer: The results and graph behavior will not exactly match with the above provided values and graph plots due to the randomization involved in our computations, but you will definitely observe optimization in every iteration.

Downloading and Running our code locally (A Backup Option) :

- ➔ If for some reason you notice that the GCP vm instance starts acting weird, then you can definitely download our project code files from the following shared drive and then run it on your local machine.
- ➔ Next, on your own local terminal, unzip and first move into the directory of the folder that you just downloaded using the "**cd**" command, and then simply use the "**ls**" to locate the setup file.
- ➔ Run the setup script to install all the required dependencies. Once the required setup is completed successfully, please use the following command to run our main script: **python3 project.py test.txt**
- ➔ Once the script runs successfully, the output will be displayed along with the plot .

For the GCP part, please make sure you remember to stop the VM instance when you're done. Also, feel free to email the following folks with any questions/concerns:
Pranav (pshirke2@jhu.edu) or Vaikunth (vaikunthd@jhu.edu).