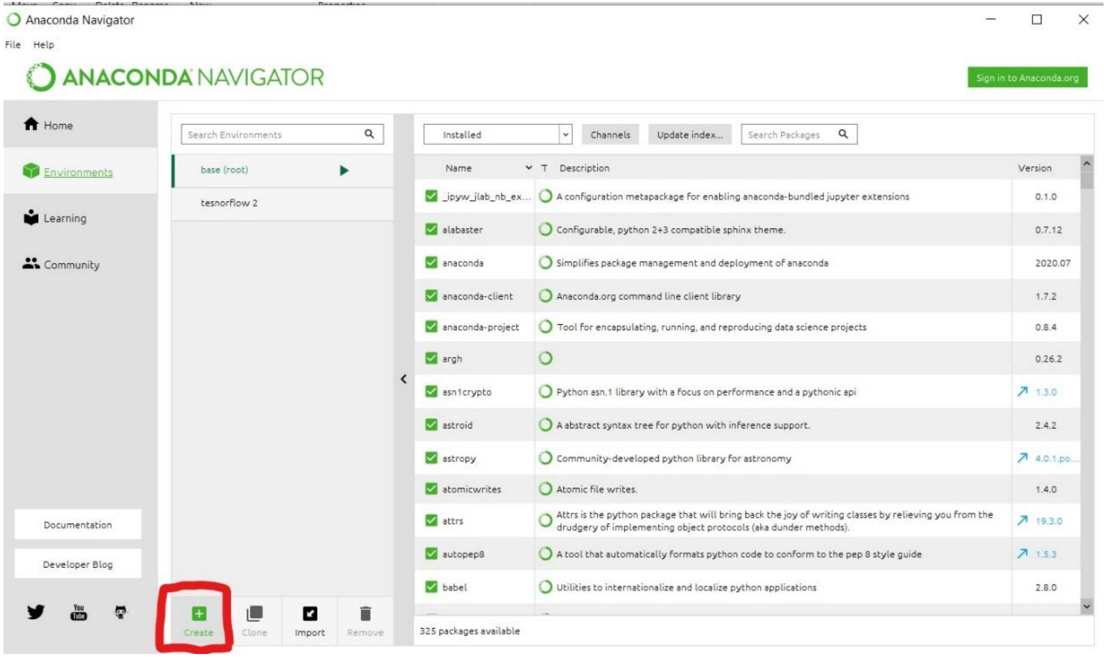
Implementation of genetic algorithm to solve hyper-parameters optimization problem

Ubuntu 20.0 operating system was used to build and run the program but it can be run in windows as well.

For both operating system anaconda is needed. Anaconda can installation process and download link can be found in:

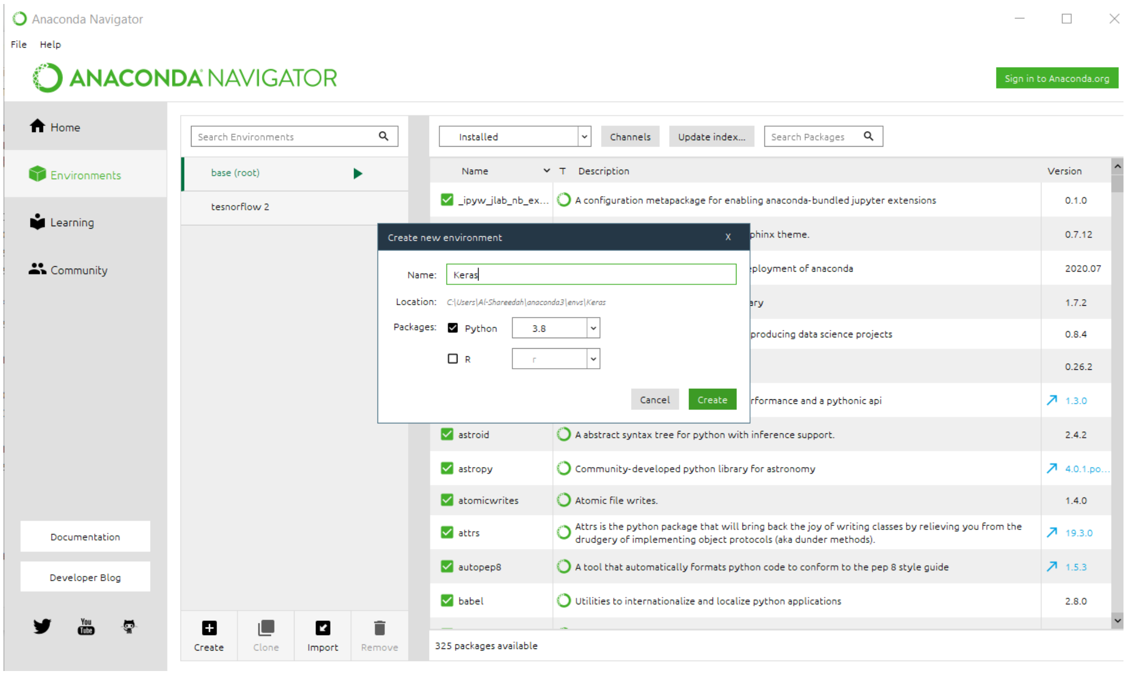
https://docs.anaconda.com/anaconda/install/

1. after installing from anaconda-navigator we create a new environment and name it anything (i.e keras).



Name	T	Description	Version
☑ _ipyw_jlab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑ alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
☑ anaconda	○	Simplifies package management and deployment of anaconda	2020.07
☑ anaconda-client	○	Anaconda.org command line client library	1.7.2
☑ anaconda-project	○	Tool for encapsulating, running, and reproducing data science projects	0.8.4
☑ argh	○		0.26.2
☑ asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	↗ 1.3.0
☑ astroid	○	A abstract syntax tree for python with inference support.	2.4.2
☑ astropy	○	Community-developed python library for astronomy	↗ 4.0.1.po...
☑ atomicwrites	○	Atomic file writes.	1.4.0
☑ attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	↗ 19.3.0
☑ autopep8	○	A tool that automatically formats python code to conform to the pep 8 style guide	↗ 1.5.3
☑ babel	○	Utilities to internationalize and localize python applications	2.8.0

325 packages available



Create new environment

Name: Keras

Location: C:\Users\Al-Shareedah\anaconda3\envs\Keras

Packages: ☑ Python 3.8

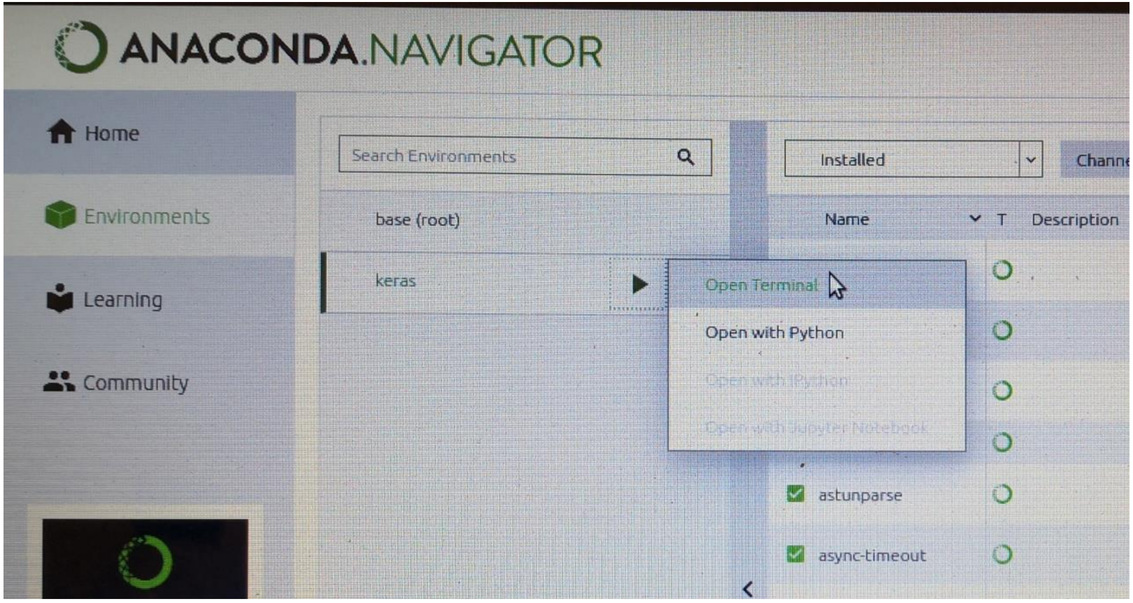
□ R r

Cancel Create

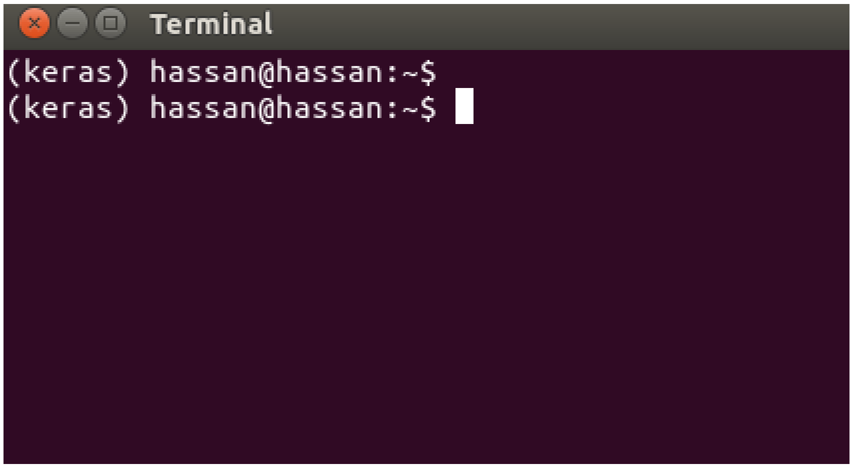
Name	T	Description	Version
☑ _ipyw_jlab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
		phinx theme.	0.7.12
		ployment of anaconda	2020.07
		ary	1.7.2
		producing data science projects	0.8.4
			0.26.2
		rformance and a pythonic api	↗ 1.3.0
☑ astroid	○	A abstract syntax tree for python with inference support.	2.4.2
☑ astropy	○	Community-developed python library for astronomy	↗ 4.0.1.po...
☑ atomicwrites	○	Atomic file writes.	1.4.0
☑ attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	↗ 19.3.0
☑ autopep8	○	A tool that automatically formats python code to conform to the pep 8 style guide	↗ 1.5.3
☑ babel	○	Utilities to internationalize and localize python applications	2.8.0

325 packages available

2. from the new environment we click run. it will prompt a menu, we select open terminal.



3. the terminal will now run the new environment, to make sure everything is correct we should see the name of the environment in bracket.



```
(keras) hassan@hassan:~$
(keras) hassan@hassan:~$
```

now that our environment is setup with python 3 and ready we start installing dependencies. the main ones that don't come by default are keras and tqdm.

4. type "`pip3 install Keras`" or "`pip install Keras`" to install keras.
5. once done, install "`pip3 install tqdm`" or "`pip install tqdm`".

now that everything is installed.

we **enter to the directory** where the Algorithms_project is located and type "`python3 main.py`"

inside the main.py file under main main function we can see the genetic algorithm configuration is set to:

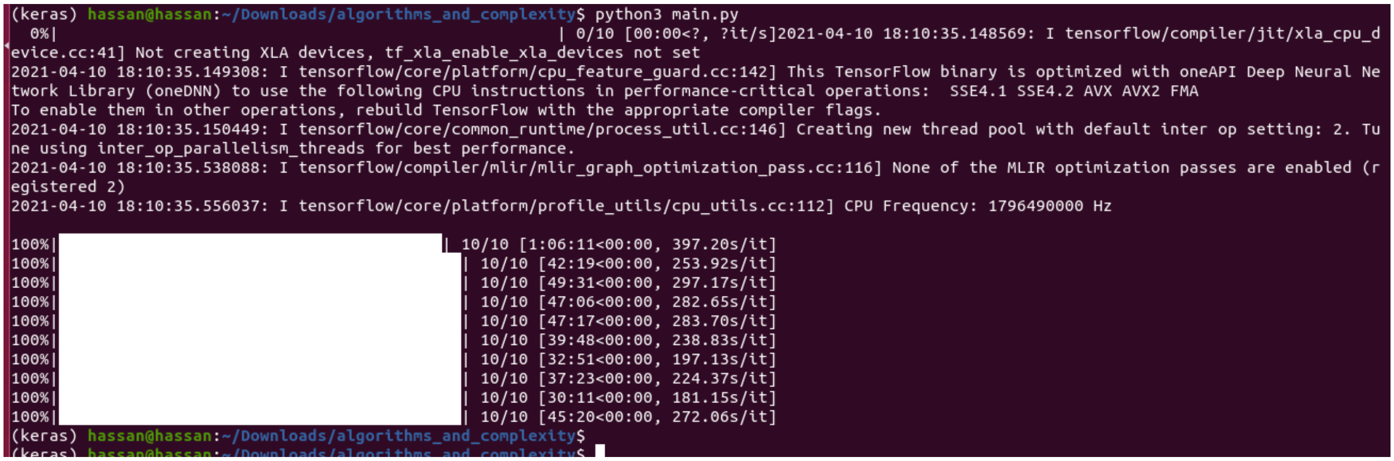
```
generations = 10  # Number of times to evole the population.

population = 10  # Number of networks in each generation.

dataset = 'mnist'
```

these can be changed to reduce the time it takes to run as it took 9 hours in my machine to complete. I would suggest to make the number of generations to 5.

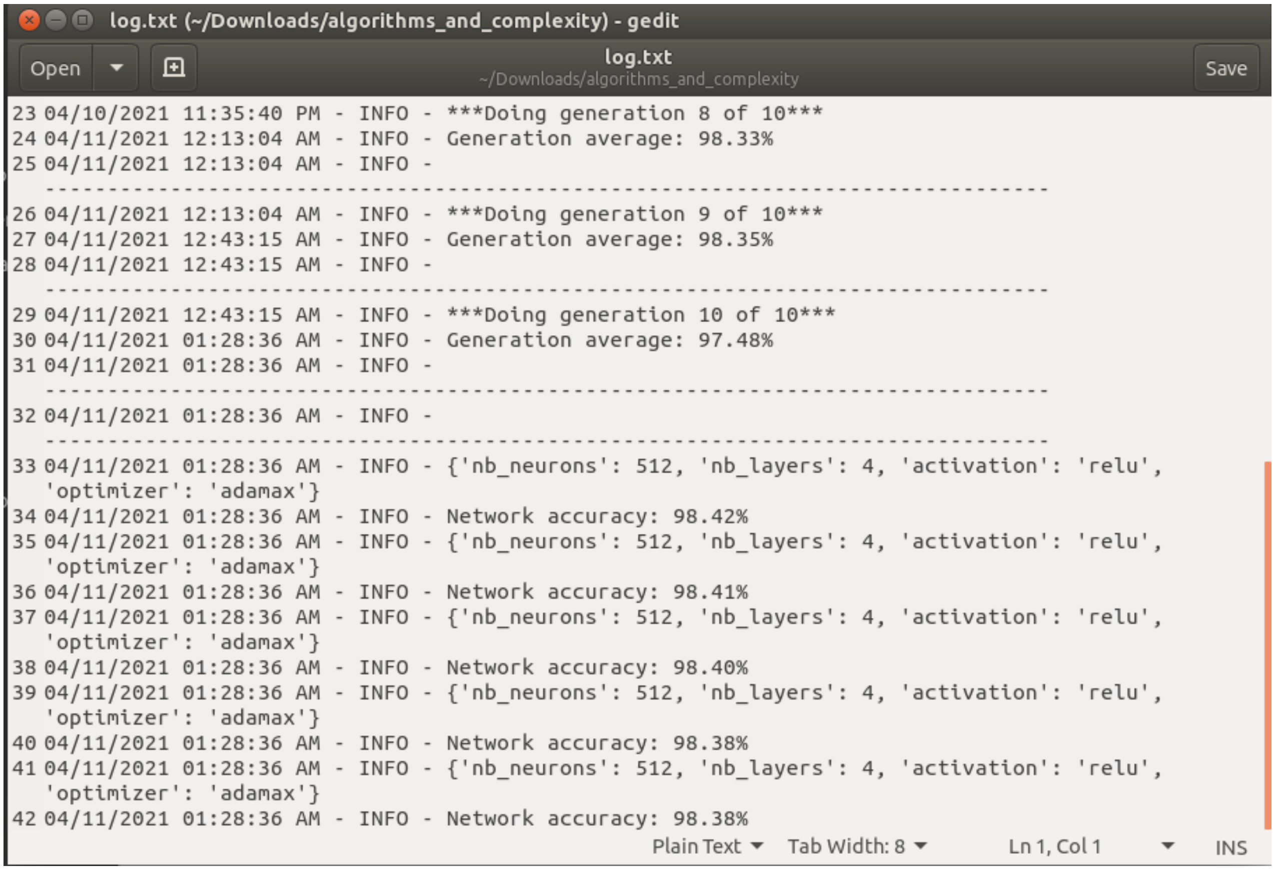
The screenshot below is how the program looks when it runs with no errors



```
(keras) hassan@hassan:~/Downloads/algorithms_and_complexity$ python3 main.py
  0%|                                                      | 0/10 [00:00<?, ?it/s]2021-04-10 18:10:35.148569: I tensorflow/compiler/jit/xla_cpu_d
evice.cc:41] Not creating XLA devices, tf_xla_enable_xla_devices not set
2021-04-10 18:10:35.149308: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Ne
twork Library (oneDNN) to use the following CPU instructions in performance-critical operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-04-10 18:10:35.150449: I tensorflow/core/common_runtime/process_util.cc:146] Creating new thread pool with default inter op setting: 2. Tu
ne using inter_op_parallelism_threads for best performance.
2021-04-10 18:10:35.538088: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled (r
egistered 2)
2021-04-10 18:10:35.556037: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 1796490000 Hz

100%|████████████████████████████████████████| 10/10 [1:06:11<00:00, 397.20s/it]
100%|██████████████████████████████████████████| 10/10 [42:19<00:00, 253.92s/it]
100%|██████████████████████████████████████████| 10/10 [49:31<00:00, 297.17s/it]
100%|██████████████████████████████████████████| 10/10 [47:06<00:00, 282.65s/it]
100%|██████████████████████████████████████████| 10/10 [47:17<00:00, 283.70s/it]
100%|██████████████████████████████████████████| 10/10 [39:48<00:00, 238.83s/it]
100%|██████████████████████████████████████████| 10/10 [32:51<00:00, 197.13s/it]
100%|██████████████████████████████████████████| 10/10 [37:23<00:00, 224.37s/it]
100%|██████████████████████████████████████████| 10/10 [30:11<00:00, 181.15s/it]
100%|██████████████████████████████████████████| 10/10 [45:20<00:00, 272.06s/it]
(keras) hassan@hassan:~/Downloads/algorithms_and_complexity$
(keras) hassan@hassan:~/Downloads/algorithms_and_complexity$ 
```

Finally, once the program finishes all the results and steps are logged in a text file called **log.txt** in the same folder as the program. The screenshot below is from the log file, we can see the top 5 networks hyper-parameter displayed at the end when all generations finished.


log.txt (~/Downloads/algorithms_and_complexity) - gedit
Open
log.txt
~/Downloads/algorithms_and_complexity
Save
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

```
04/10/2021 11:35:40 PM - INFO - ***Doing generation 8 of 10***
04/11/2021 12:13:04 AM - INFO - Generation average: 98.33%
04/11/2021 12:13:04 AM - INFO -
 --------------------------------------------------------------------------------
04/11/2021 12:13:04 AM - INFO - ***Doing generation 9 of 10***
04/11/2021 12:43:15 AM - INFO - Generation average: 98.35%
04/11/2021 12:43:15 AM - INFO -
 --------------------------------------------------------------------------------
04/11/2021 12:43:15 AM - INFO - ***Doing generation 10 of 10***
04/11/2021 01:28:36 AM - INFO - Generation average: 97.48%
04/11/2021 01:28:36 AM - INFO -
 --------------------------------------------------------------------------------
04/11/2021 01:28:36 AM - INFO -
 --------------------------------------------------------------------------------
04/11/2021 01:28:36 AM - INFO - {'nb_neurons': 512, 'nb_layers': 4, 'activation': 'relu',
 'optimizer': 'adamax'}
04/11/2021 01:28:36 AM - INFO - Network accuracy: 98.42%
04/11/2021 01:28:36 AM - INFO - {'nb_neurons': 512, 'nb_layers': 4, 'activation': 'relu',
 'optimizer': 'adamax'}
04/11/2021 01:28:36 AM - INFO - Network accuracy: 98.41%
04/11/2021 01:28:36 AM - INFO - {'nb_neurons': 512, 'nb_layers': 4, 'activation': 'relu',
 'optimizer': 'adamax'}
04/11/2021 01:28:36 AM - INFO - Network accuracy: 98.40%
04/11/2021 01:28:36 AM - INFO - {'nb_neurons': 512, 'nb_layers': 4, 'activation': 'relu',
 'optimizer': 'adamax'}
04/11/2021 01:28:36 AM - INFO - Network accuracy: 98.38%
04/11/2021 01:28:36 AM - INFO - {'nb_neurons': 512, 'nb_layers': 4, 'activation': 'relu',
 'optimizer': 'adamax'}
04/11/2021 01:28:36 AM - INFO - Network accuracy: 98.38%
```