Configuration Manual

MSc Research Project
MSc in cloud computing

Ashish Ekhande

Student ID: x18148221

School of Computing
National College of Ireland

Supervisor: Manuel Tova-Izquierdo

Student Name:	Ashish Ekhande
Student ID:	x18148221
Programme:	MSc in cloud computing
Year:	2018
Module:	MSc Research Project
Supervisor:	Manuel Tova-Izquierdo
Submission Due Date:	20/12/2018
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	8

Signature:	
Date:	23rd April 2020

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Ashish Ekhande

x18148221

1 Introduction

Various tools and libraries have been used to assemble the project. This document is aimed to give information about the procedure to follow to execute the project.

2 Prerequisites

As I have coded my project in python programming language, user of the document has to have understanding of python programming language and libraries associated in document.

Default recommended version of python is 3.7 and can be downloaded from official website. On linux operating system, python v2 comes as default. Python 3.7 can be installed using command line.

3 Required Python Libraries

Implementation of this project requires python libraries. To install additional libraries user has to install pip for python version 3.7. If pip is not getting installed on the system, updating the packages may solve the issue.Libraries that have to be installed are listed below:

3.1 Install pip

sudo apt install python3-pip

Figure 1: pip installation

3.2 Tensorflow 2

If user does not want to install virtual environment then he has to specify –user while executing the command. This command will install or update the tensorflow library to latest version.

pip3 install –user –no-cache-dir –upgrade tensorflow

Figure 2: Tensorflow 2

3.3 sklearn

sudo apt-get install python3-sklearn python3-sklearn-lib All essential libraries are covered

Figure 3: Install skrearn

when user installs tensorflow and sklearn packages. However, in case of error you may have to install matplotlib library as well. Library can be installed with command:

python3 -m pip install –upgrade matplotlib

4 Implementation steps and analysis

Once user has executed all the commands specified in section 3, he can execute the project. In order to do that, he has to run 2 applications using python3.

- python3 simple_test_app.py

- python3 composite_test_app.py

When first file is executed, it will trigger the Simple scaling controller. Simple scaling controller will process the inputs without passing them through linear regression. As inputs are passed to ANN without rounding them of with CSC, complexity of this model is more.

python3 simple_test_app.py

```
ashish@ubuntu:~/5_auto_scaling$ python3 simple_test_app.py
[0 0 1 2 2 3 3 5 4]
(9,)
2020-04-23 09:13:55.612080: W tensorflow/stream_executor/platform/default/d
uch file or directory
2020-04-23 09:13:55.620683: W tensorflow/stream_executor/platform/default/d
ect file: No such file or directory
2020-04-23 09:13:55.620740: W tensorflow/compiler/tf2tensorrt/utils/py_util
ibraries mentioned above are installed properly.
2020-04-23 09:13:57.438870: W tensorflow/stream_executor/platform/default/d
le or directory
2020-04-23 09:13:57.439021: E tensorflow/stream_executor/cuda/cuda_driver.c
2020-04-23 09:13:57.439084: I tensorflow/stream_executor/cuda/cuda_diagnost
2020-04-23 09:13:57.440790: I tensorflow/core/platform/cpu_feature_guard.cc
2020-04-23 09:13:57.478962: I tensorflow/core/platform/profile_utils/cpu_ut
2020-04-23 09:13:57.479998: I tensorflow/compiler/xla/service/service.cc:16
2020-04-23 09:13:57.480037: I tensorflow/compiler/xla/service/service.cc:17
Train on 400 samples
Epoch 1/200
400/400 [==============================] - 1s 1ms/sample - loss: 214.5736
Epoch 2/200
400/400 [==============================] - 0s 28us/sample - loss: 37.1484
Epoch 3/200
400/400 [==============================] - 0s 31us/sample - loss: 33.1175
Epoch 4/200
400/400 [==============================] - 0s 75us/sample - loss: 20.8587
Epoch 5/200
400/400 [==============================] - 0s 94us/sample - loss: 10.1403
Epoch 6/200
400/400 [==============================] - 0s 75us/sample - loss: 5.9170
Epoch 7/200
400/400 [==============================] - 0s 51us/sample - loss: 3.8133
Epoch 8/200
400/400 [==============================] - 0s 69us/sample - loss: 2.4470
Epoch 9/200
400/400 [==============================] - 0s 65us/sample - loss: 2.0046
Epoch 10/200
400/400 [==============================] - 0s 57us/sample - loss: 1.6843
Epoch 11/200
400/400 [==============================] - 0s 43us/sample - loss: 1.5146
Epoch 12/200
400/400 [==============================] - 0s 58us/sample - loss: 1.4043
Epoch 13/200
400/400 [                              ] - 0s 58us/sample - loss: 1.3436
```

Figure 4: Epoch execution in system

When simple test app is executed, data is being generated randomly by task generator. Epoch shows that data set is being learnt from model, Then from this data, task queue is generated.

```
(1, 9)
generated 0 task inside the queue
no task to process
generated 3 task inside the queue
3 task processed by the machine
generated 5 task inside the queue
3 task processed by the machine
generated 6 task inside the queue
3 task processed by the machine
generated 7 task inside the queue
3 task processed by the machine
generated 10 task inside the queue
3 task processed by the machine
generated 11 task inside the queue
3 task processed by the machine
generated 11 task inside the queue
3 task processed by the machine
generated 13 task inside the queue
3 task processed by the machine
generated 15 task inside the queue
3 task processed by the machine
generated 15 task inside the queue
3 task processed by the machine
generated 16 task inside the queue
3 task processed by the machine
generated 17 task inside the queue
the vale of data is [ 4  7  5 10  6 11  7 11  8 13  9 15 10 15 11 16 12 17]
(9, 2)
(1, 9)
the shape of X is  (1, 9)
[[17.474983]]
17 task processed by the machine
generated 19 task inside the queue
the vale of data is [ 5 10  6 11  7 11  8 13  9 15 10 15 11 16 12 17 13 19]
(9, 2)
(1, 9)
```

Figure 5: Task queue

As tasks are processed new tasks are generated by task generator. Result shows that accuracy is maintained in this system.
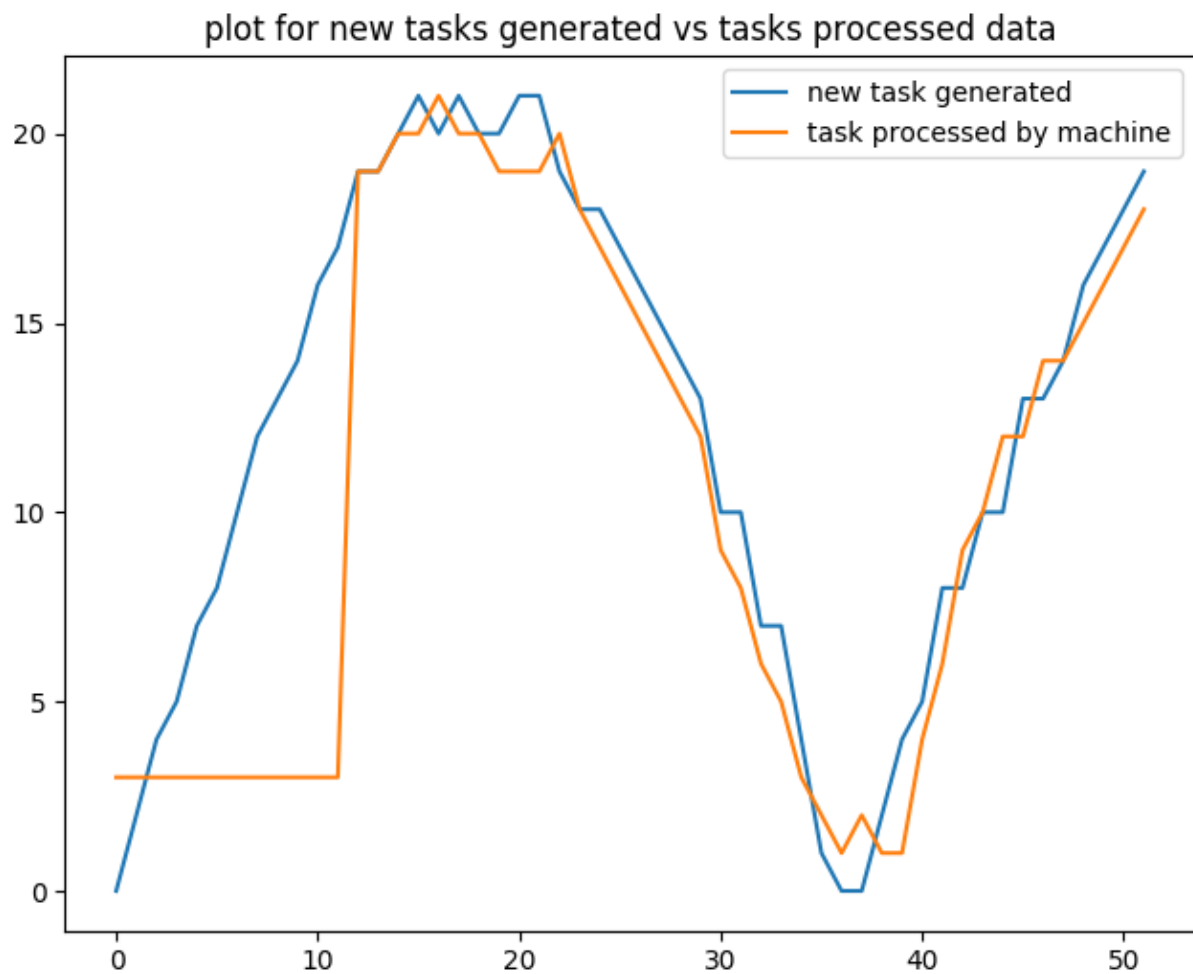
Figure 6: SSC result

4.1 The proposed CSC

In the next step,tasks generated are first passed through linear regression. Input generated is then feeded to ANN. This leads to less complexity while maintaining considerable accuracy.

python3 composite_test_app.py

```
ashish@ubuntu:~/auto_scaling_v1$ python3 composite_test_app.py
2020-04-09 06:35:05.900857: W tensorflow/stream_executor/platform/default/ds
uch file or directory
2020-04-09 06:35:05.902401: W tensorflow/stream_executor/platform/default/ds
ect file: No such file or directory
2020-04-09 06:35:05.902479: W tensorflow/compiler/tf2tensorrt/utils/py_utils
ibraries mentioned above are installed properly.
2020-04-09 06:35:07.661959: W tensorflow/stream_executor/platform/default/ds
le or directory
2020-04-09 06:35:07.662139: E tensorflow/stream_executor/cuda/cuda_driver.cc
2020-04-09 06:35:07.662263: I tensorflow/stream_executor/cuda/cuda_diagnosti
2020-04-09 06:35:07.663401: I tensorflow/core/platform/cpu_feature_guard.cc:
2020-04-09 06:35:07.699499: I tensorflow/core/platform/profile_utils/cpu_uti
2020-04-09 06:35:07.700624: I tensorflow/compiler/xla/service/service.cc:168
2020-04-09 06:35:07.700696: I tensorflow/compiler/xla/service/service.cc:176
Train on 300 samples
Epoch 1/200
300/300 [==============================] - 1s 2ms/sample - loss: 189.6874
Epoch 2/200
300/300 [==============================] - 0s 165us/sample - loss: 165.3007
Epoch 3/200
300/300 [==============================] - 0s 143us/sample - loss: 101.3384
Epoch 4/200
300/300 [==============================] - 0s 106us/sample - loss: 63.6727
Epoch 5/200
300/300 [==============================] - 0s 182us/sample - loss: 51.2088
Epoch 6/200
300/300 [==============================] - 0s 88us/sample - loss: 44.6430
Epoch 7/200
300/300 [==============================] - 0s 134us/sample - loss: 41.4755
Epoch 8/200
300/300 [==============================] - 0s 90us/sample - loss: 37.8568
Epoch 9/200
300/300 [==============================] - 0s 174us/sample - loss: 35.6349
Epoch 10/200
300/300 [==============================] - 0s 147us/sample - loss: 32.8532
Epoch 11/200
300/300 [==============================] - 0s 134us/sample - loss: 30.9878
Epoch 12/200
300/300 [==============================] - 0s 158us/sample - loss: 28.5279
Epoch 13/200
300/300 [==============================] - 0s 87us/sample - loss: 27.7659
```
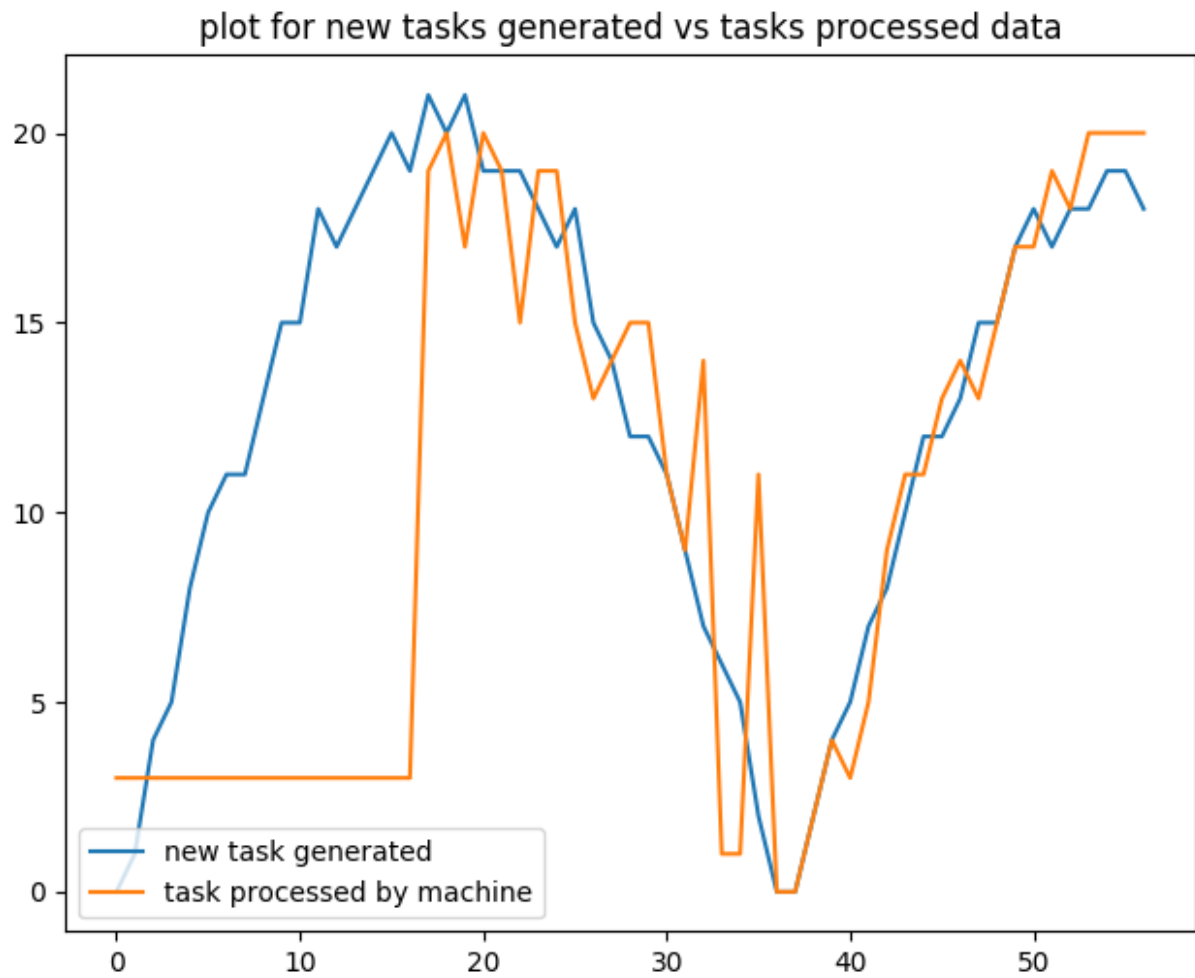
Figure 7: Composite epoch

Figure 8: CSC result

References