CSYE 7374 **Cognitive Computing and Deep Neural Network**

Documentation Report for Assignment -2:

Lab 1: Facebook Messenger Chatbot:

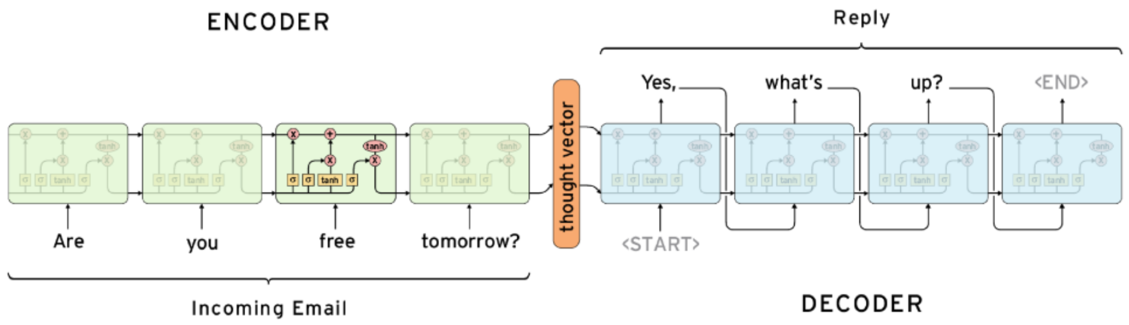
Aim: -

To create a Facebook Messenger chatbot using seq2seq model which is trained on the Facebook chats from one's own past data.

Theory: -

Chatbots are "computer programs which conduct conversation through auditory or textual methods". Chatbots that use deep learning are almost all using some variant of a sequence to sequence (Seq2Seq) model.

A sequence to sequence model is composed of 2 main components, an encoder RNN and a decoder RNN. From a high level, the encoder's job is to encapsulate the information of the input text into a fixed representation. The decoder's is to take that representation, and generate a variable length text that best responds to it.

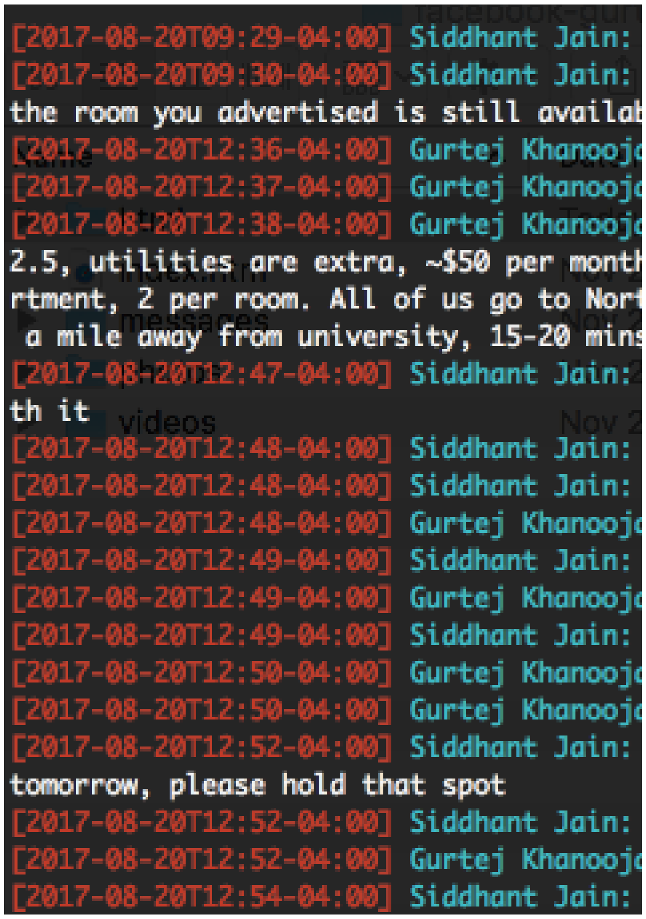


Procedure: -

Step -1 Dataset Selection

Since we wanted to make the chatbot reply the way we talk, the first step was to parse the Facebooks messenger chats from the past.

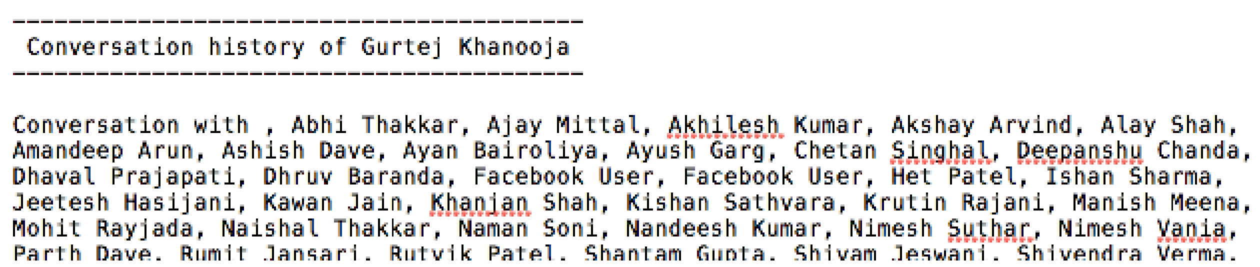
Screenshot of parsing data:



```
[2017-08-20T09:29-04:00] Siddhant Jain:
[2017-08-20T09:30-04:00] Siddhant Jain:
the room you advertised is still availab
[2017-08-20T12:36-04:00] Gurtej Khanooja
[2017-08-20T12:37-04:00] Gurtej Khanooja
[2017-08-20T12:38-04:00] Gurtej Khanooja
2.5, utilities are extra, ~$50 per month
rtment, 2 per room. All of us go to Nort
 a mile away from university, 15-20 mins
[2017-08-20T12:47-04:00] Siddhant Jain:
th it
[2017-08-20T12:48-04:00] Siddhant Jain:
[2017-08-20T12:48-04:00] Siddhant Jain:
[2017-08-20T12:48-04:00] Gurtej Khanooja
[2017-08-20T12:49-04:00] Siddhant Jain:
[2017-08-20T12:49-04:00] Gurtej Khanooja
[2017-08-20T12:49-04:00] Siddhant Jain:
[2017-08-20T12:50-04:00] Gurtej Khanooja
[2017-08-20T12:50-04:00] Gurtej Khanooja
[2017-08-20T12:52-04:00] Siddhant Jain:
tomorrow, please hold that spot
[2017-08-20T12:52-04:00] Siddhant Jain:
[2017-08-20T12:52-04:00] Gurtej Khanooja
[2017-08-20T12:54-04:00] Siddhant Jain:
```

Parsed text file of the chats:

`

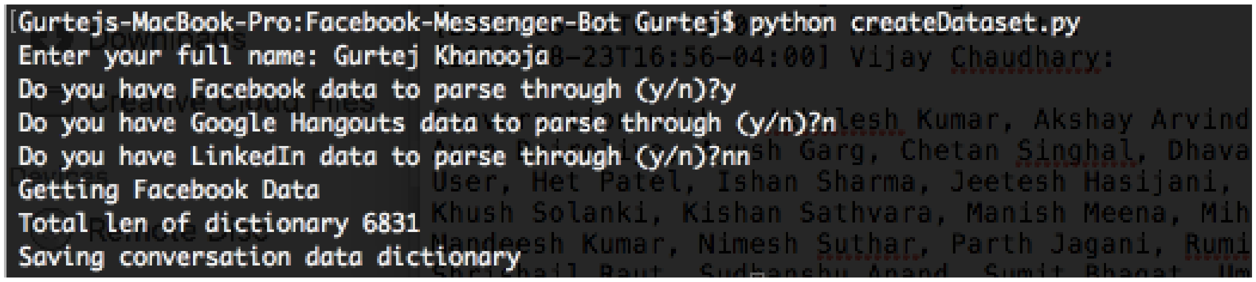


```
------------------------------------------
 Conversation history of Gurtej Khanooja
------------------------------------------

Conversation with , Abhi Thakkar, Ajay Mittal, Akhilesh Kumar, Akshay Arvind, Alay Shah,
Amandeep Arun, Ashish Dave, Ayan Bairoliya, Ayush Garg, Chetan Singhal, Deepanshu Chanda,
Dhaval Prajapati, Dhruv Baranda, Facebook User, Facebook User, Het Patel, Ishan Sharma,
Jeetesh Hasijani, Kawan Jain, Khanjan Shah, Kishan Sathvara, Krutin Rajani, Manish Meena,
Mohit Rayjada, Naishal Thakkar, Naman Soni, Nandeesh Kumar, Nimesh Suthar, Nimesh Vania,
Parth Dave, Rumit Jansari, Rutvik Patel, Shantam Gupta, Shivam Jeswani, Shivendra Verma,
```

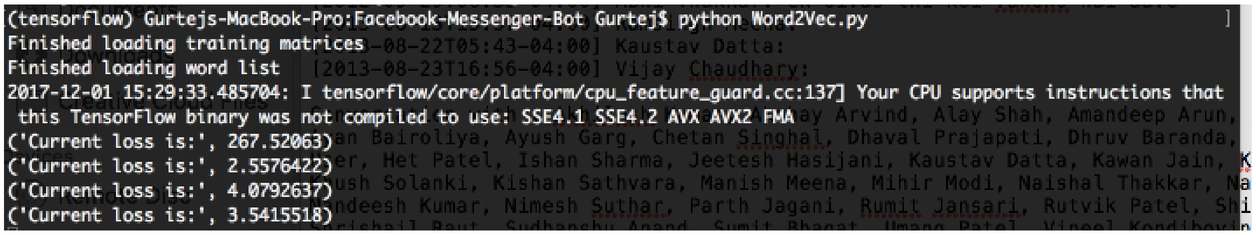
Step -2 Using createdataset.py function on parsed data

This script will create a file named conversationDictionary.npy which is a Numpy object that contains pairs in the form of (FRIENDS_MESSAGE, YOUR RESPONSE). A file named conversationData.txt will also be created. This is simply a large text file the dictionary data in a unified form.



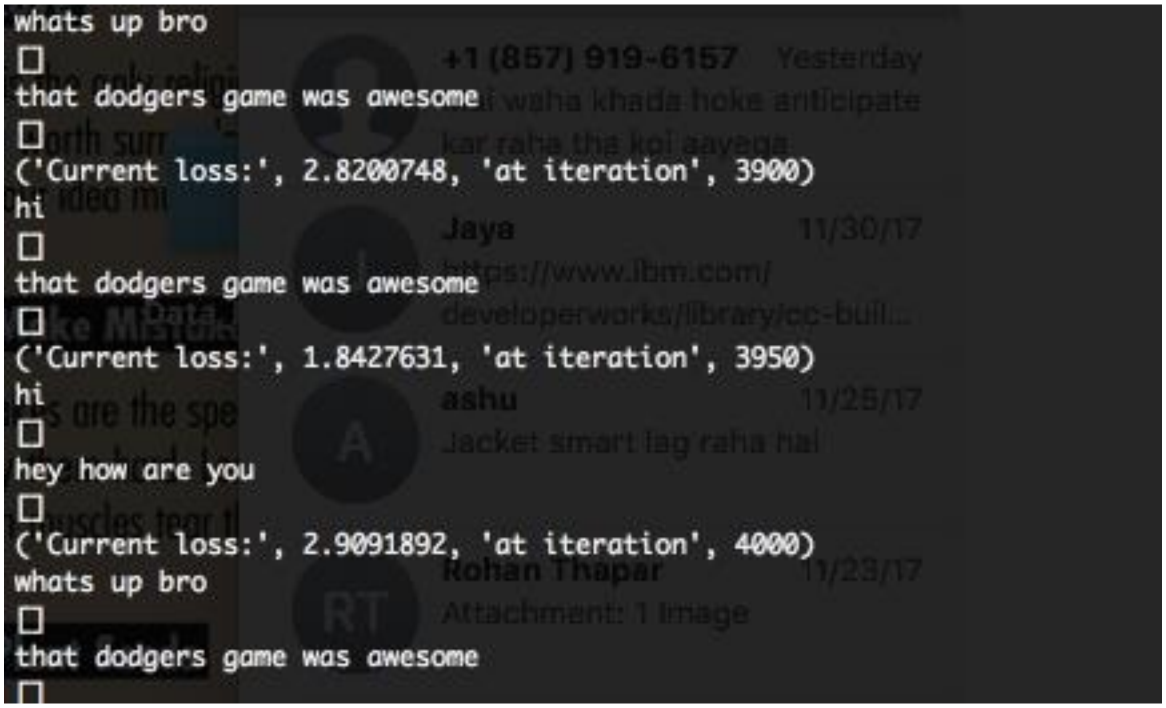
```
Gurtejs-MacBook-Pro:Facebook-Messenger-Bot Gurtej$ python createDataset.py
Enter your full name: Gurtej Khanooja
Do you have Facebook data to parse through (y/n)?y
Do you have Google Hangouts data to parse through (y/n)?n
Do you have LinkedIn data to parse through (y/n)?nn
Getting Facebook Data
Total len of dictionary 6831
Saving conversation data dictionary
```

Step -3 Using Word2Vec.py function



```
(tensorflow) Gurtejs-MacBook-Pro:Facebook-Messenger-Bot Gurtej$ python Word2Vec.py
Finished loading training matrices
Finished loading word list
2017-12-01 15:29:33.485704: I tensorflow/core/platform/cpu_feature_guard.cc:137] Your CPU supports instructions that
 this TensorFlow binary was not compiled to use: SSE4.1 SSE4.2 AVX AVX2 FMA
('Current loss is:', 267.52063)
('Current loss is:', 2.5576422)
('Current loss is:', 4.0792637)
('Current loss is:', 3.5415518)
```

Step -4 Using Seq2Seq.py function (creating seq2seq model)

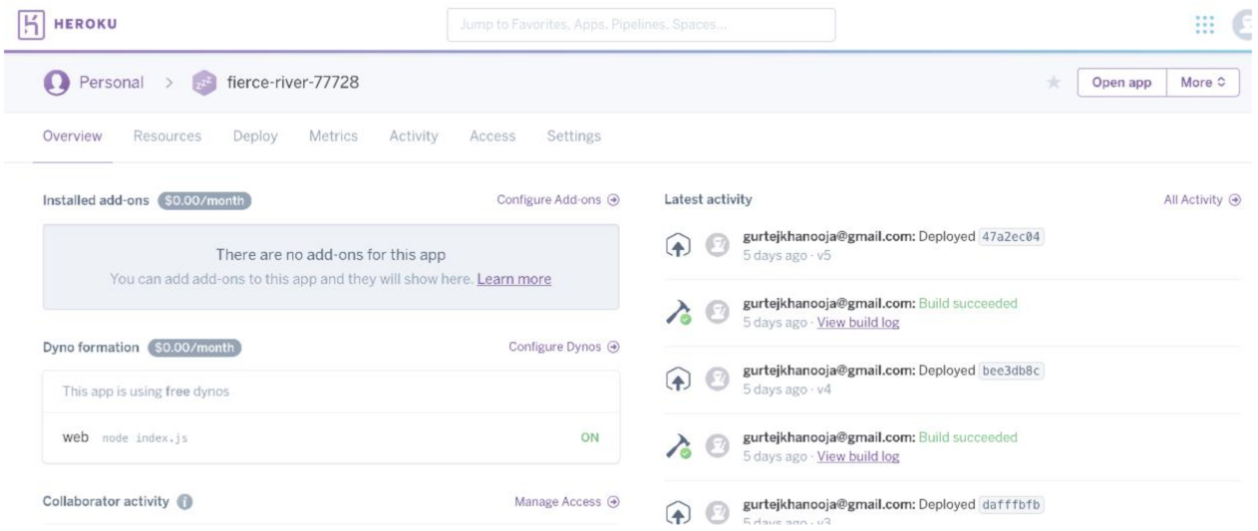


```
whats up bro
□
that dodgers game was awesome
□
('Current loss:', 2.8200748, 'at iteration', 3900)
hi
□
that dodgers game was awesome
□
('Current loss:', 1.8427631, 'at iteration', 3950)
hi
□
hey how are you
□
('Current loss:', 2.9091892, 'at iteration', 4000)
whats up bro
□
that dodgers game was awesome
□
```

The model here doesn't seem to learn anything here as the number of iterations are very less for the model to learn anything useful from the limited data which we provided it. Earlier, I ran the model for 98500 iterations and model had started picking the words which I use in my conversation but my computer crashed and I cannot use that screen shot now. But my final model of messenger app is deployed using the earlier model.

Step -5 Creating a messenger app

- Build the server, and host on Heroku:



Installed add-ons $0.00/month

There are no add-ons for this app
You can add add-ons to this app and they will show here. Learn more

Dyno formation $0.00/month

This app is using free dynos

web `node index.js` ON

Collaborator activity

Latest activity

gurtejkhanooja@gmail.com: Deployed 47a2ec04
5 days ago · v5

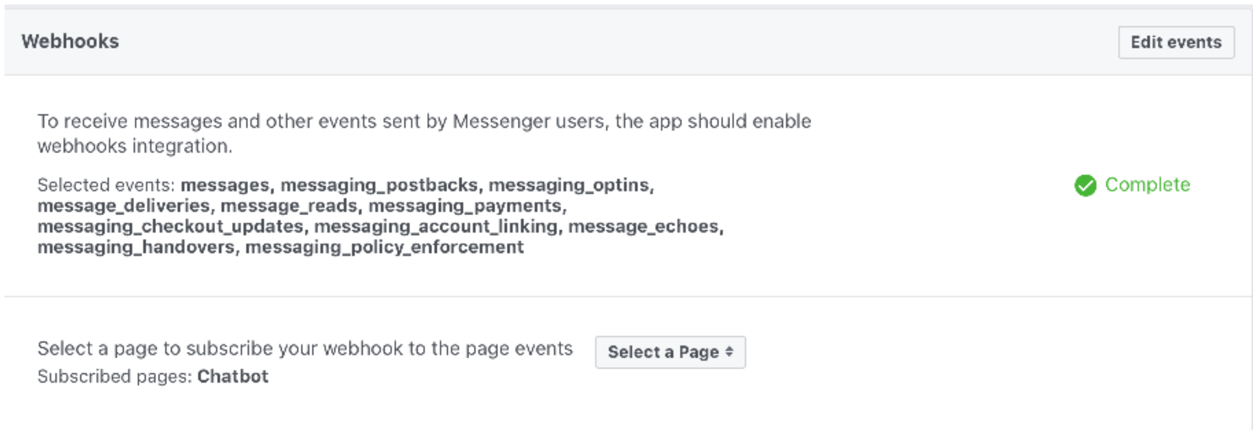
gurtejkhanooja@gmail.com: Build succeeded
5 days ago · View build log

gurtejkhanooja@gmail.com: Deployed bee3db8c
5 days ago · v4

gurtejkhanooja@gmail.com: Build succeeded
5 days ago · View build log

gurtejkhanooja@gmail.com: Deployed dafffbfb
5 days ago · v3

- Create a Facebook App/Page, set up the webhook, get page token, and trigger the app.



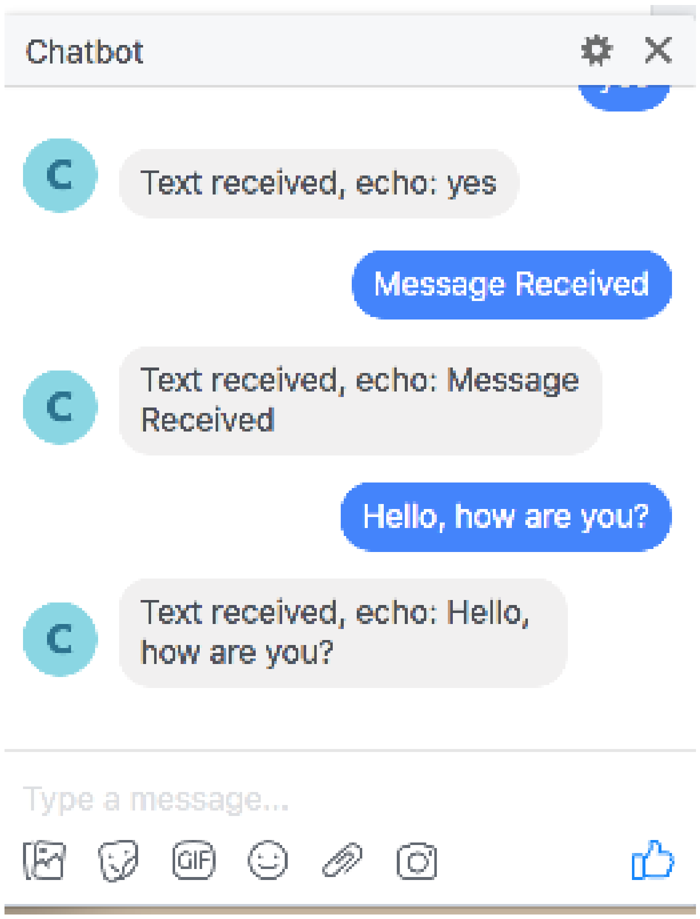
To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Selected events: **messages, messaging_postbacks, messaging_optins, message_deliveries, message_reads, messaging_payments, messaging_checkout_updates, messaging_account_linking, message_echoes, messaging_handovers, messaging_policy_enforcement**

Complete

Select a page to subscribe your webhook to the page events
Subscribed pages: **Chatbot**

- Add an API endpoint to **index.js** so that the bot can respond with messages. (The app should start echoing the messages)



Text received, echo: yes

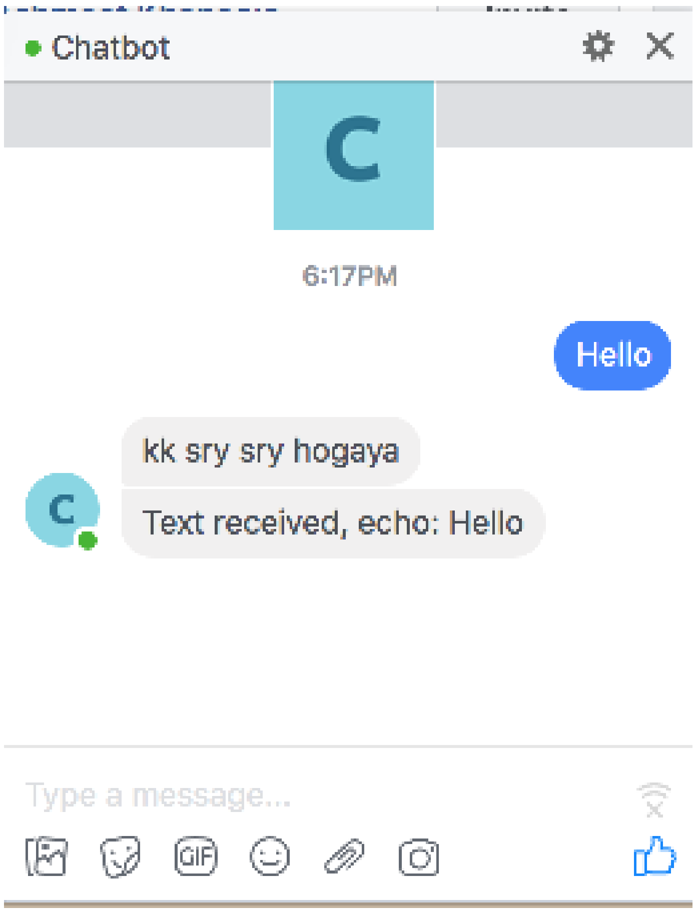
Message Received

Text received, echo: Message Received

Hello, how are you?

Text received, echo: Hello, how are you?

Step -6 Deploying using flask server and integrating model through express app


Chatbot

6:17PM

Hello

kk sry sry hogaya

Text received, echo: Hello

Lab 2: Natural Machine Translation:

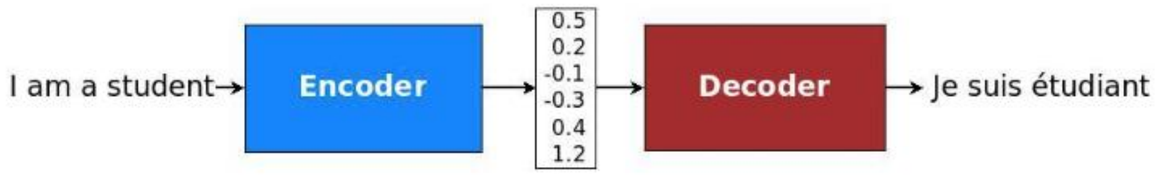
Aim:

To use Seq2Seq modeling for natural language translation i.e. to translate one language to another language.

Theory: -

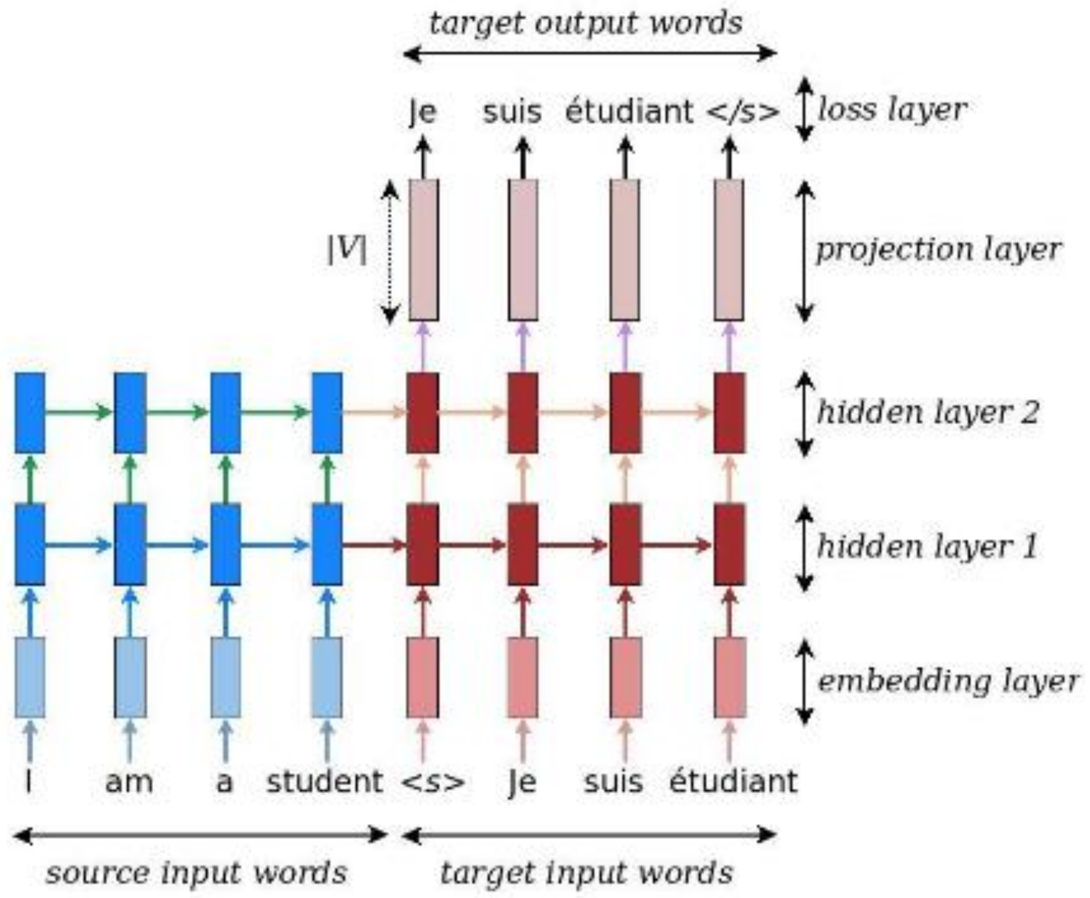
Back in the old days, traditional phrase-based translation systems performed their task by breaking up source sentences into multiple chunks and then translated them phrase-by-phrase. This led to disfluency in the translation outputs and was not quite like how we, humans, translate. We read the entire source sentence, understand its meaning, and then produce a translation. Neural Machine Translation (NMT) mimics that!

Specifically, an NMT system first reads the source sentence using an *encoder* to build a "thought" vector, a sequence of numbers that represents the sentence meaning; a *decoder*, then, processes the sentence vector to emit a translation, as illustrated in Figure below. This is often referred to as the *encoder-decoder architecture*. In this manner, NMT addresses the local translation problem in the traditional phrase-based approach: it can capture *long-range dependencies* in languages, e.g., gender agreements; syntax structures; etc., and produce much more fluent translations

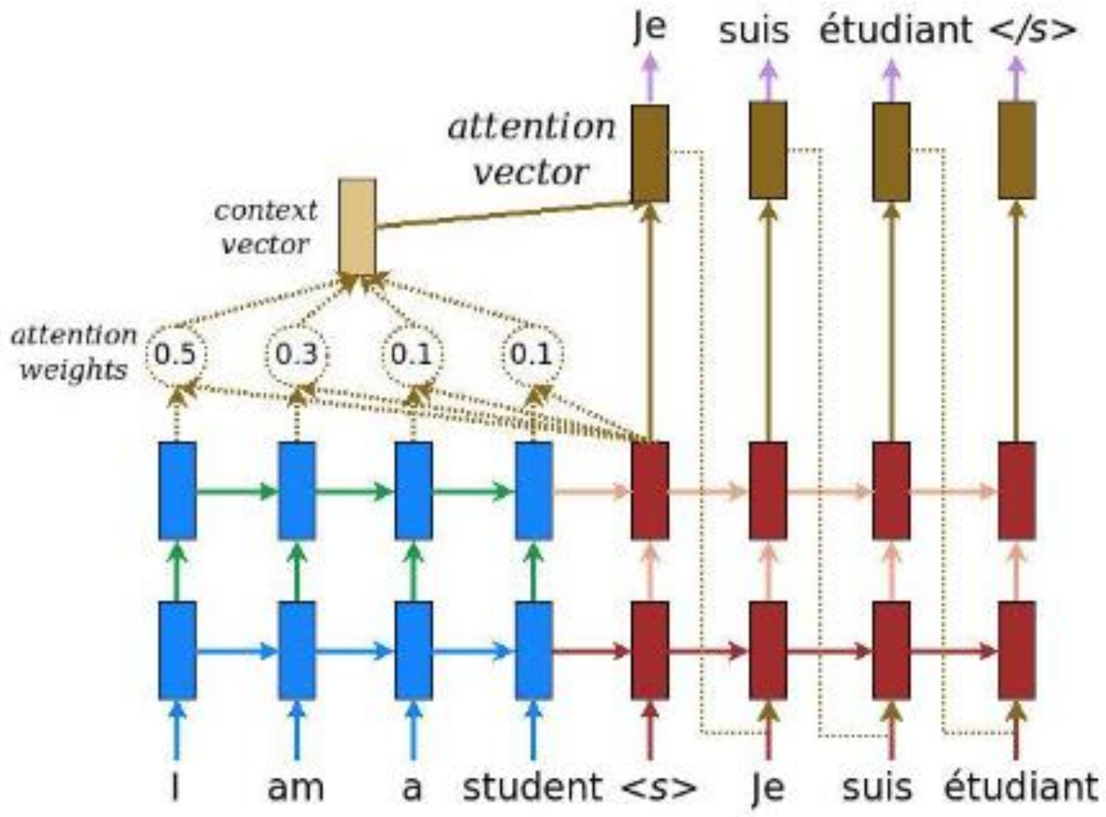


In this tutorial, we consider as examples a *deep multi-layer RNN* which is unidirectional and uses LSTM as a recurrent unit. We show an example of such a model in Figure below.

`



We train NMT model with attention mechanism, which has been used in several state-of-the-art systems. The figure below shows NMT with attention mechanism:



Procedure: -

- NMT model without attention mechanism

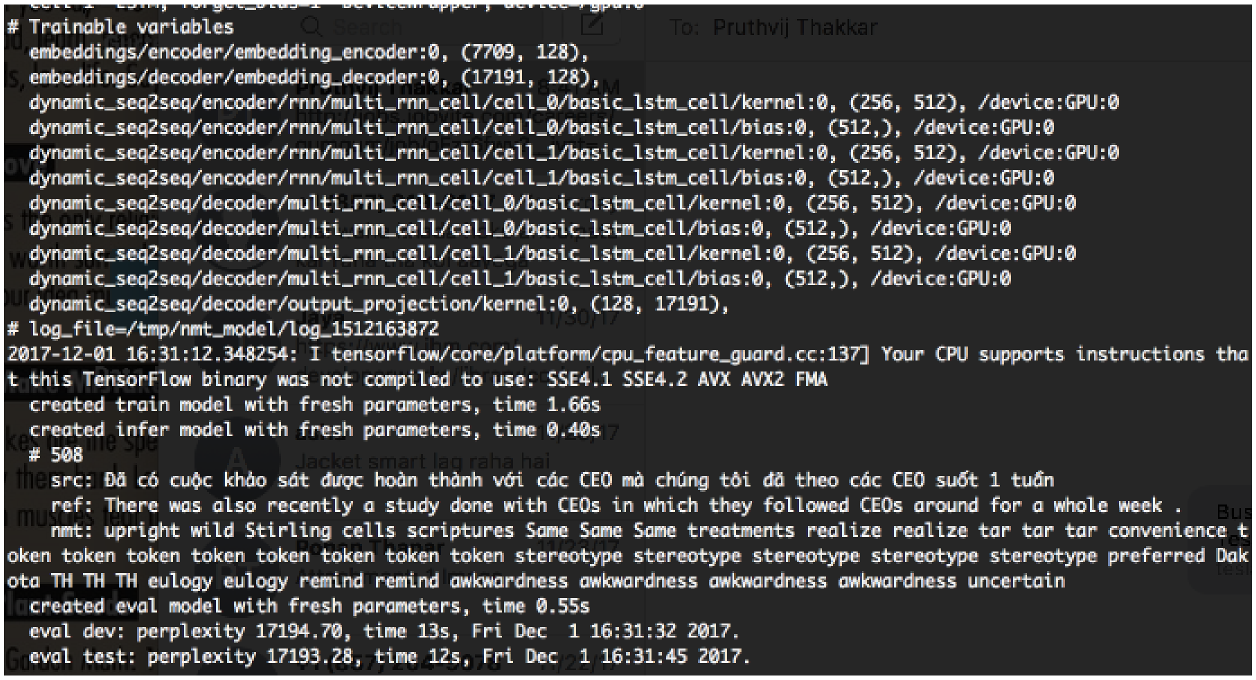
  Training the model:

  Command used:

```
mkdir /tmp/nmt_model

python -m nmt.nmt \
--src=vi --tgt=en \
--vocab_prefix=/tmp/nmt_data/vocab  \
--train_prefix=/tmp/nmt_data/train \
--dev_prefix=/tmp/nmt_data/tst2012  \
--test_prefix=/tmp/nmt_data/tst2013 \
--out_dir=/tmp/nmt_model \
--num_train_steps=12000 \
--steps_per_stats=100 \
--num_layers=2 \
--num_units=128 \
--dropout=0.2 \
--metrics=bleu
```

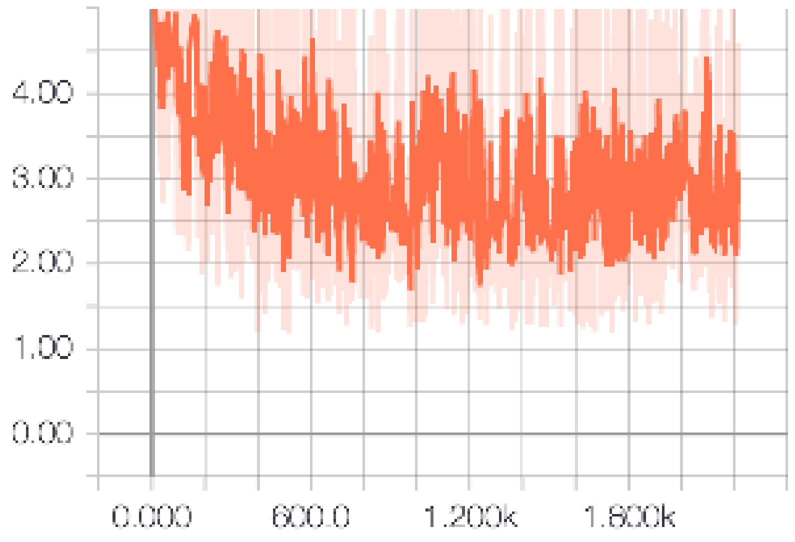
Snapshot of tranning in the terminal:



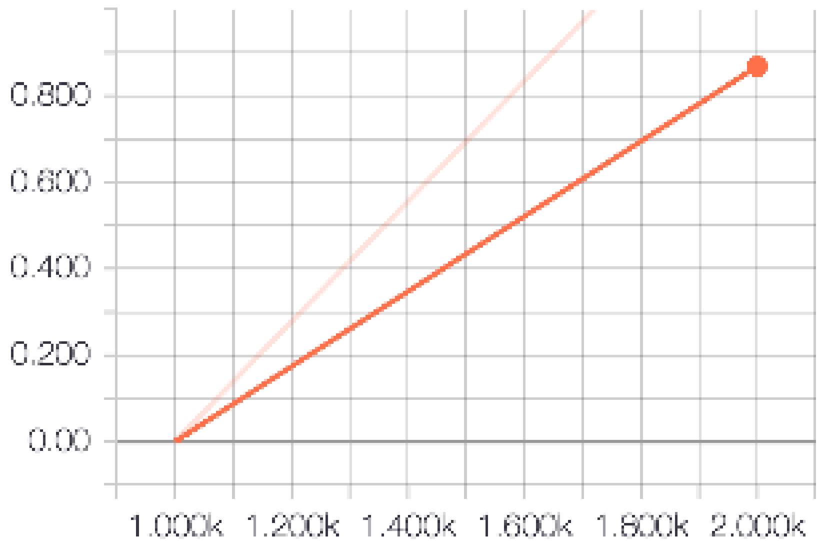
```
# Trainable variables
  embeddings/encoder/embedding_encoder:0, (7709, 128),
  embeddings/decoder/embedding_decoder:0, (17191, 128),
  dynamic_seq2seq/encoder/rnn/multi_rnn_cell/cell_0/basic_lstm_cell/kernel:0, (256, 512), /device:GPU:0
  dynamic_seq2seq/encoder/rnn/multi_rnn_cell/cell_0/basic_lstm_cell/bias:0, (512,), /device:GPU:0
  dynamic_seq2seq/encoder/rnn/multi_rnn_cell/cell_1/basic_lstm_cell/kernel:0, (256, 512), /device:GPU:0
  dynamic_seq2seq/encoder/rnn/multi_rnn_cell/cell_1/basic_lstm_cell/bias:0, (512,), /device:GPU:0
  dynamic_seq2seq/decoder/multi_rnn_cell/cell_0/basic_lstm_cell/kernel:0, (256, 512), /device:GPU:0
  dynamic_seq2seq/decoder/multi_rnn_cell/cell_0/basic_lstm_cell/bias:0, (512,), /device:GPU:0
  dynamic_seq2seq/decoder/multi_rnn_cell/cell_1/basic_lstm_cell/kernel:0, (256, 512), /device:GPU:0
  dynamic_seq2seq/decoder/multi_rnn_cell/cell_1/basic_lstm_cell/bias:0, (512,), /device:GPU:0
  dynamic_seq2seq/decoder/output_projection/kernel:0, (128, 17191),
# log_file=/tmp/nmt_model/log_1512163872
2017-12-01 16:31:12.348254: I tensorflow/core/platform/cpu_feature_guard.cc:137] Your CPU supports instructions tha
t this TensorFlow binary was not compiled to use: SSE4.1 SSE4.2 AVX AVX2 FMA
  created train model with fresh parameters, time 1.66s
  created infer model with fresh parameters, time 0.40s
  # 508
    src: Đã có cuộc khảo sát được hoàn thành với các CEO mà chúng tôi đã theo các CEO suốt 1 tuần
    ref: There was also recently a study done with CEOs in which they followed CEOs around for a whole week .
    nmt: upright wild Stirling cells scriptures Same Same Same treatments realize realize tar tar tar convenience t
oken token token token token token token token stereotype stereotype stereotype stereotype stereotype preferred Dak
ota TH TH TH eulogy eulogy remind remind awkwardness awkwardness awkwardness awkwardness uncertain
  created eval model with fresh parameters, time 0.55s
  eval dev: perplexity 17194.70, time 13s, Fri Dec  1 16:31:32 2017.
  eval test: perplexity 17193.28, time 12s, Fri Dec  1 16:31:45 2017.
```

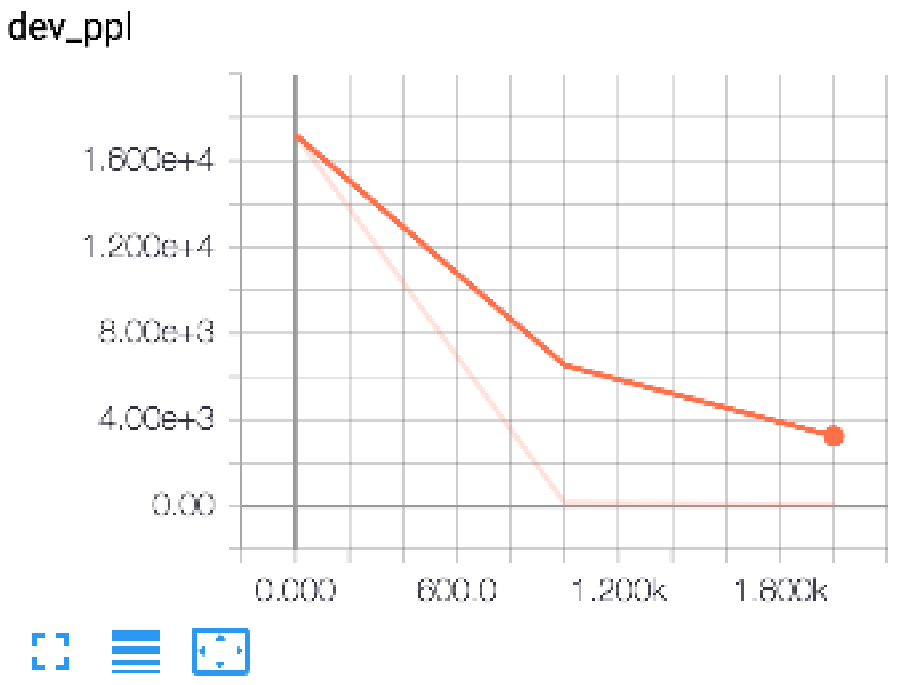
Tensorbord tranning graphs:

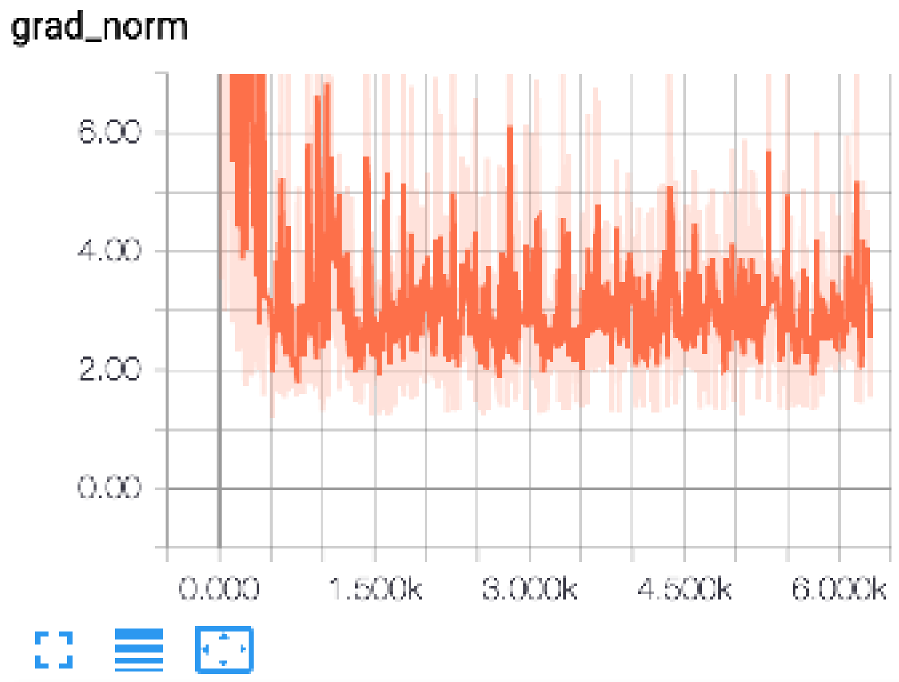
clipped_gradient



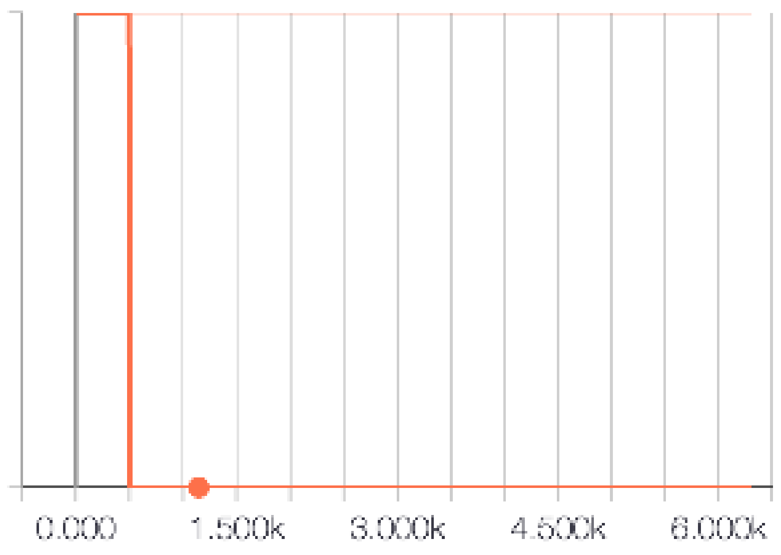
dev_bleu



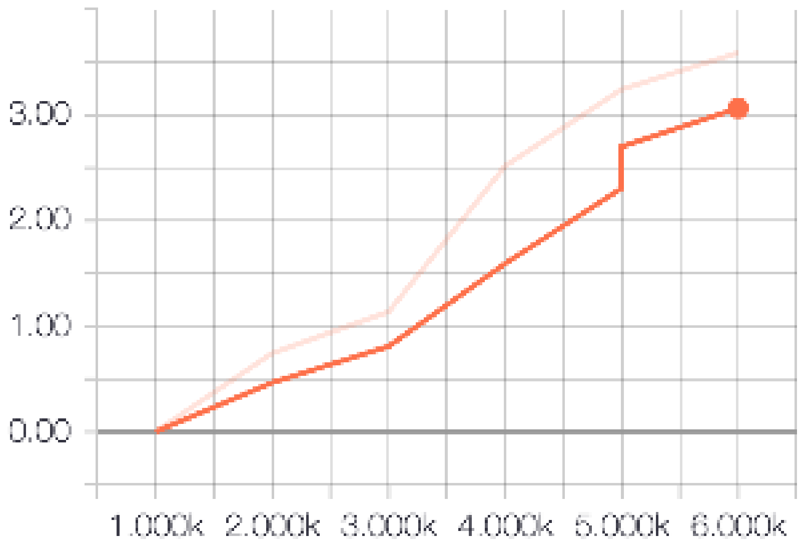

dev_ppl
1.600e+4
1.200e+4
8.00e+3
4.00e+3
0.00
0.000
600.0
1.200k
1.800k


grad_norm
6.00
4.00
2.00
0.00
0.000
1.500k
3.000k
4.500k
6.000k

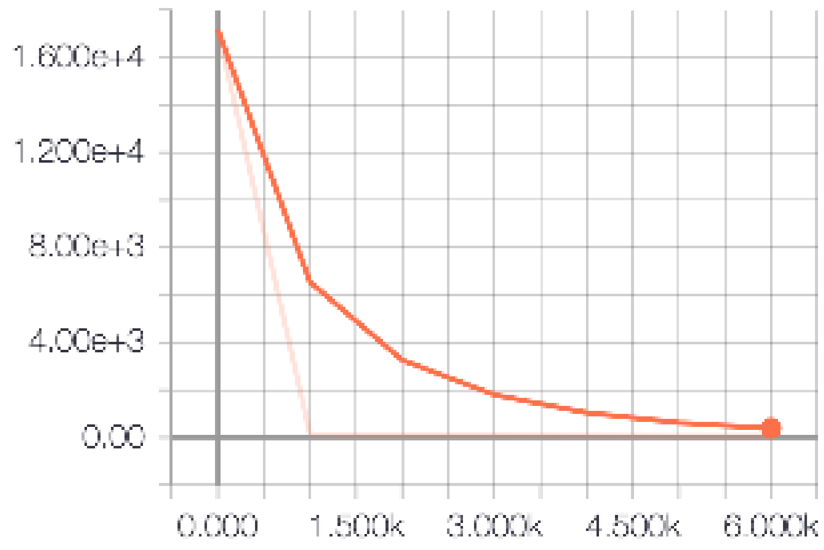
lr_1



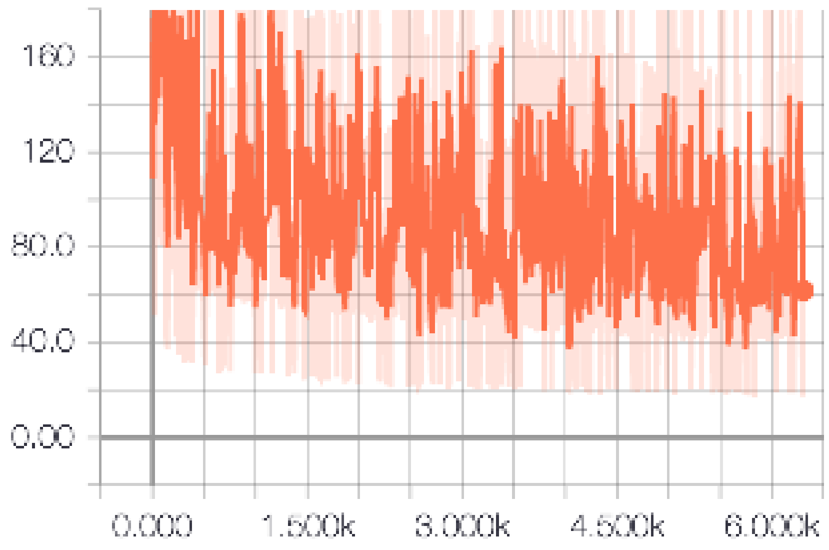
test_bleu



test_ppl



train_loss



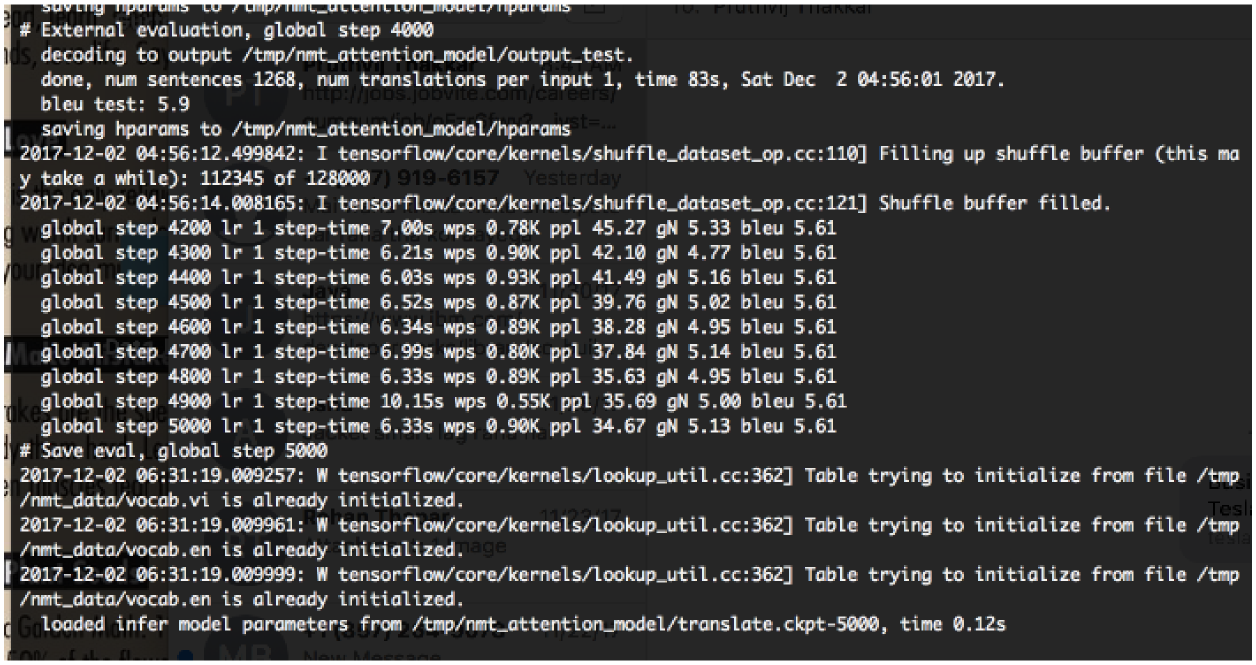
- NMT model with attention mechanism:

  Command used:

  ```
  mkdir /tmp/nmt_attention_model
  ```

```
      python -m nmt.nmt \
--attention=scaled_luong \
--src=vi --tgt=en \
--vocab_prefix=/tmp/nmt_data/vocab  \
--train_prefix=/tmp/nmt_data/train \
--dev_prefix=/tmp/nmt_data/tst2012  \
--test_prefix=/tmp/nmt_data/tst2013 \
--out_dir=/tmp/nmt_attention_model \
--num_train_steps=12000 \
--steps_per_stats=100 \
--num_layers=2 \
--num_units=128 \
--dropout=0.2 \
--metrics=bleu
```

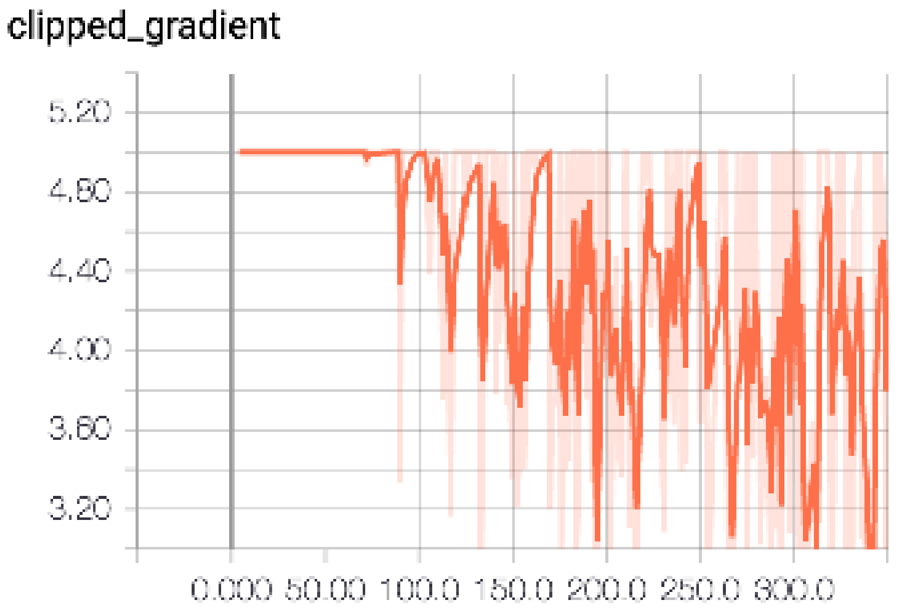
Snapshot of tranning in the terminal:

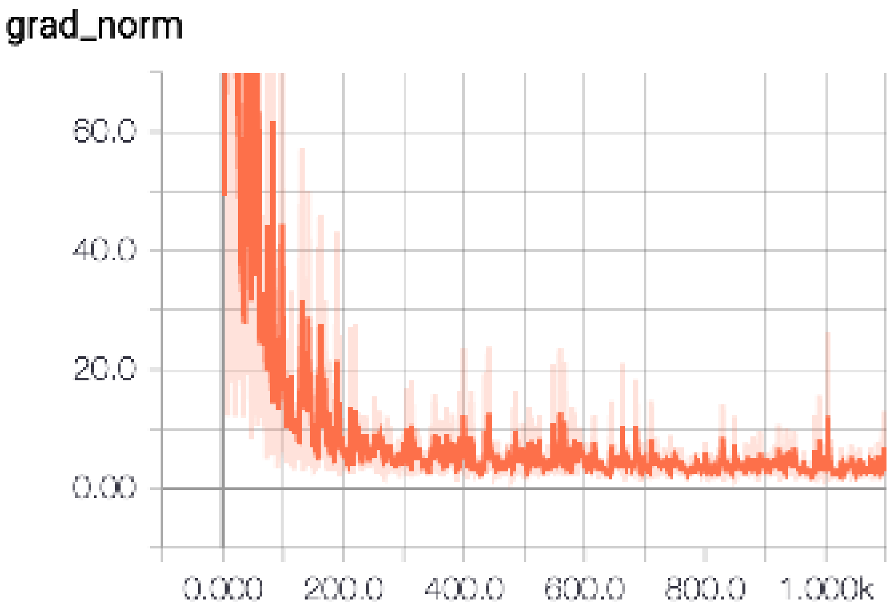


```
  saving hparams to /tmp/nmt_attention_model/hparams
# External evaluation, global step 4000
  decoding to output /tmp/nmt_attention_model/output_test.
  done, num sentences 1268, num translations per input 1, time 83s, Sat Dec  2 04:56:01 2017.
  bleu test: 5.9
  saving hparams to /tmp/nmt_attention_model/hparams
2017-12-02 04:56:12.499842: I tensorflow/core/kernels/shuffle_dataset_op.cc:110] Filling up shuffle buffer (this ma
y take a while): 112345 of 128000
2017-12-02 04:56:14.008165: I tensorflow/core/kernels/shuffle_dataset_op.cc:121] Shuffle buffer filled.
  global step 4200 lr 1 step-time 7.00s wps 0.78K ppl 45.27 gN 5.33 bleu 5.61
  global step 4300 lr 1 step-time 6.21s wps 0.90K ppl 42.10 gN 4.77 bleu 5.61
  global step 4400 lr 1 step-time 6.03s wps 0.93K ppl 41.49 gN 5.16 bleu 5.61
  global step 4500 lr 1 step-time 6.52s wps 0.87K ppl 39.76 gN 5.02 bleu 5.61
  global step 4600 lr 1 step-time 6.34s wps 0.89K ppl 38.28 gN 4.95 bleu 5.61
  global step 4700 lr 1 step-time 6.99s wps 0.80K ppl 37.84 gN 5.14 bleu 5.61
  global step 4800 lr 1 step-time 6.33s wps 0.89K ppl 35.63 gN 4.95 bleu 5.61
  global step 4900 lr 1 step-time 10.15s wps 0.55K ppl 35.69 gN 5.00 bleu 5.61
  global step 5000 lr 1 step-time 6.33s wps 0.90K ppl 34.67 gN 5.13 bleu 5.61
# Save eval, global step 5000
2017-12-02 06:31:19.009257: W tensorflow/core/kernels/lookup_util.cc:362] Table trying to initialize from file /tmp
/nmt_data/vocab.vi is already initialized.
2017-12-02 06:31:19.009961: W tensorflow/core/kernels/lookup_util.cc:362] Table trying to initialize from file /tmp
/nmt_data/vocab.en is already initialized.
2017-12-02 06:31:19.009999: W tensorflow/core/kernels/lookup_util.cc:362] Table trying to initialize from file /tmp
/nmt_data/vocab.en is already initialized.
  loaded infer model parameters from /tmp/nmt_attention_model/translate.ckpt-5000, time 0.12s
```

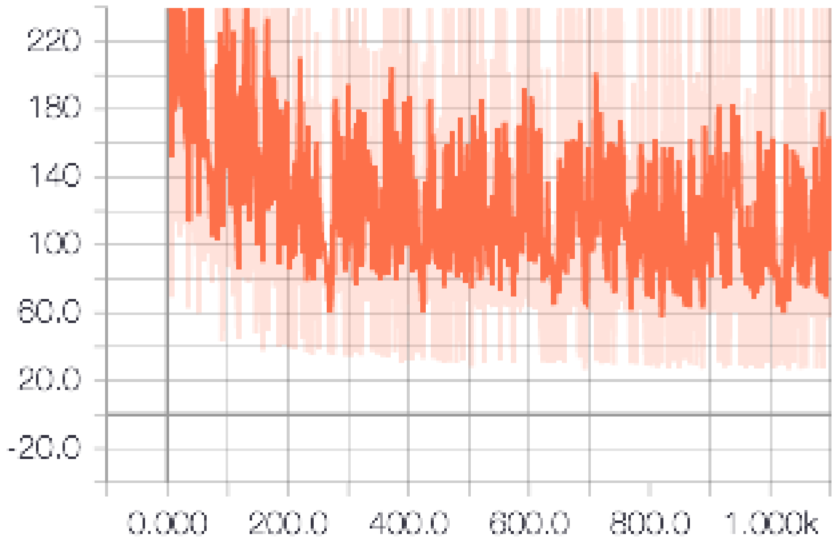
Tensorbord training graphs:

There are all different graphs generated in tensorbord as illustrated by the model above but I am showing only two graphs for NMT model with attention mechanism:


clipped_gradient
5.20
4.80
4.40
4.00
3.60
3.20
0.000
50.00
100.0
150.0
200.0
250.0
300.0


grad_norm
60.0
40.0
20.0
0.00
0.000
200.0
400.0
600.0
800.0
1.000k

train_loss



Lab 3: Bonsai Reinforcement Learning:

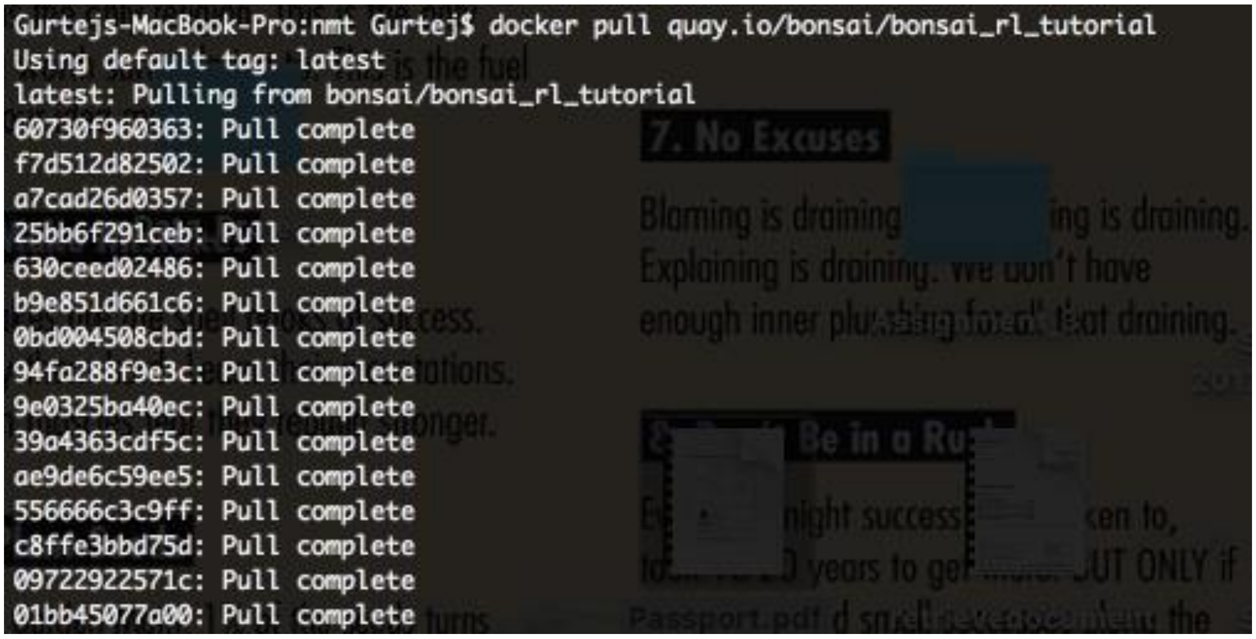
<u>Aim: -</u>

To run famous reinforcement learning examples of cartpole and lunar lander and see the effect by changing the reward function.

<u>Procedure: -</u>

Cartpole:

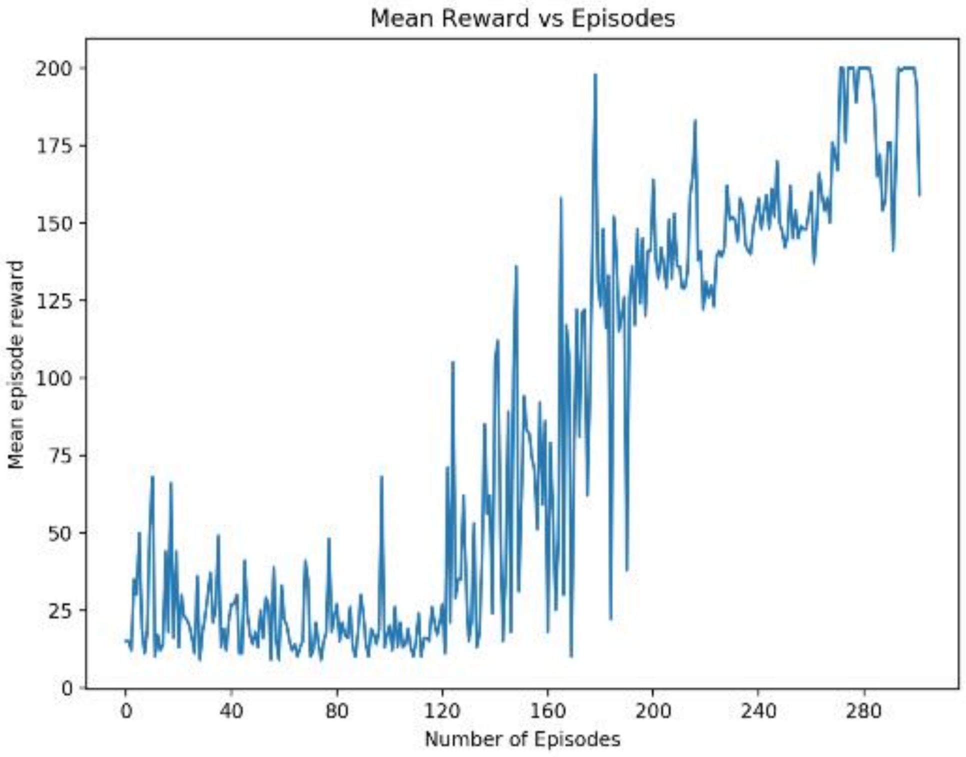
- Downloading docker images:



```
Gurtejs-MacBook-Pro:nmt Gurtej$ docker pull quay.io/bonsai/bonsai_rl_tutorial
Using default tag: latest
latest: Pulling from bonsai/bonsai_rl_tutorial
60730f960363: Pull complete
f7d512d82502: Pull complete
a7cad26d0357: Pull complete
25bb6f291ceb: Pull complete
630ceed02486: Pull complete
b9e851d661c6: Pull complete
0bd004508cbd: Pull complete
94fa288f9e3c: Pull complete
9e0325ba40ec: Pull complete
39a4363cdf5c: Pull complete
ae9de6c59ee5: Pull complete
556666c3c9ff: Pull complete
c8ffe3bbd75d: Pull complete
09722922571c: Pull complete
01bb45077a00: Pull complete
```

- Cartpole with following reward function:

```
if not done:
   reward = 1.0
elif self.steps_beyond_done is None:
   # Pole just fell!
   self.steps_beyond_done = 0
   reward = 1.0
```



```
[2017-12-03 01:32:21,575] Starting new video recorder writing to /bonsai/baselines/tutorials/cartpole/openaigym
.video.0.bonsai.video000296.mp4
[2017-12-03 01:32:24,307] Starting new video recorder writing to /bonsai/baselines/tutorials/cartpole/openaigym
.video.0.bonsai.video000297.mp4
[2017-12-03 01:32:27,216] Starting new video recorder writing to /bonsai/baselines/tutorials/cartpole/openaigym
.video.0.bonsai.video000298.mp4
[2017-12-03 01:32:30,092] Starting new video recorder writing to /bonsai/baselines/tutorials/cartpole/openaigym
.video.0.bonsai.video000299.mp4
```

```
--------------------------------------
| % time spent exploring  | 2        |
| episodes                | 300      |
| mean 100 episode reward | 159      |
| steps                   | 24846    |
--------------------------------------
```

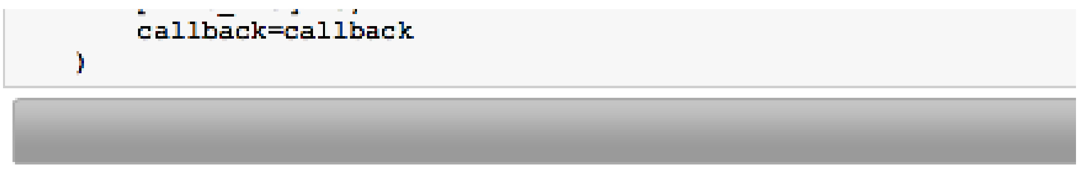
```
[2017-12-03 01:32:32,970] Starting new video recorder writing to /bonsai/baselines/tutorials/cartpole/openaigym
.video.0.bonsai.video000300.mp4
[2017-12-03 01:32:35,740] Starting new video recorder writing to /bonsai/baselines/tutorials/cartpole/openaigym
.video.0.bonsai.video000301.mp4
[2017-12-03 01:32:38,421] Starting new video recorder writing to /bonsai/baselines/tutorials/cartpole/openaigym
.video.0.bonsai.video000302.mp4
```

```
Restored model with mean reward: 142.1
INFO:tensorflow:Restoring parameters from /tmp/tmpmq9fnodf/model
```

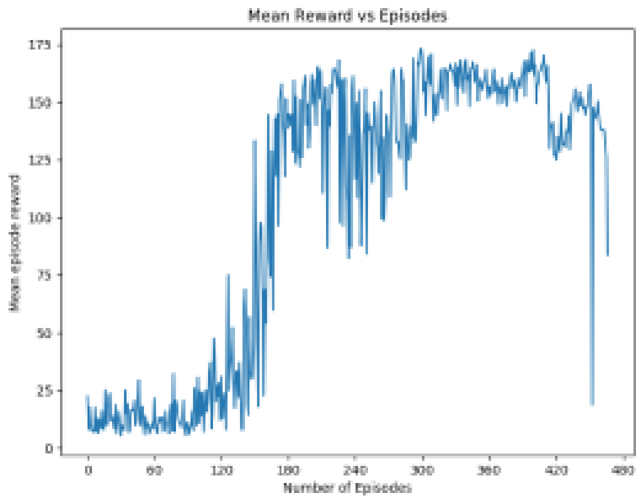
```
[2017-12-03 01:32:38,585] Restoring parameters from /tmp/tmpmq9fnodf/model
[2017-12-03 01:32:38,681] Finished writing results. You can upload them to the scoreboard via gym.upload('/bons
ai/baselines/tutorials/cartpole')
```

- Cartpole with following reward function:

```
if not done:
  reward = (1-math.pow(abs(x)/
self.x_threshold, 0.4))*0.5+(1-
math.pow(abs(theta)/
self.theta_threshold_radians,
0.4))*0.5
elif self.steps_beyond_done is None:
  # Pole just fell!
  self.steps_beyond_done = 0
  reward = 0.0
```

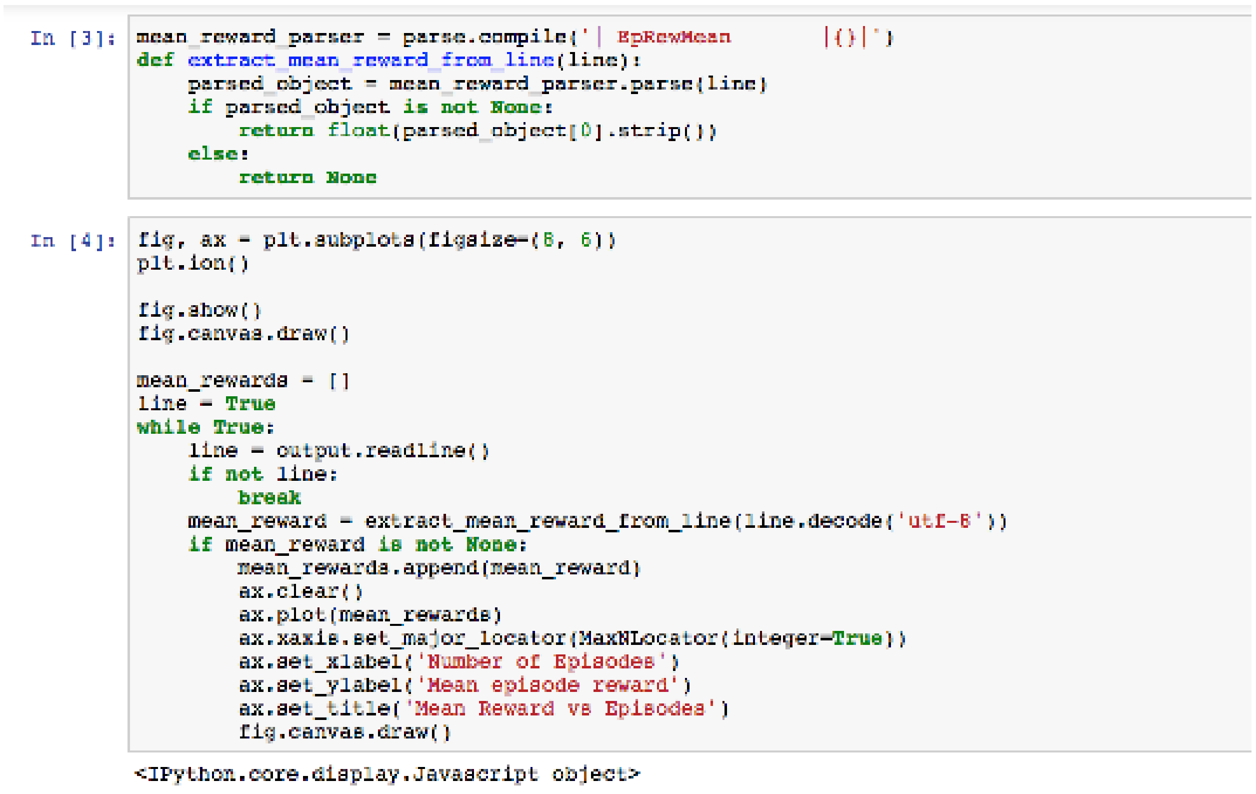


```
        callback=callback
    )
```



- Lunar Lander

Mean Reward vs Episode



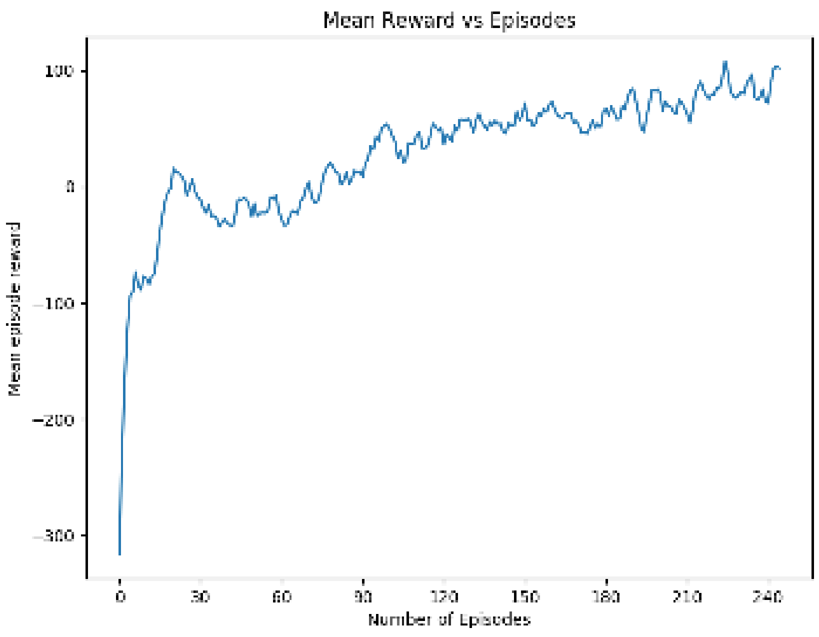
```
In [3]: mean_reward_parser = parse.compile('| EpRewMean       |{}|')
        def extract_mean_reward_from_line(line):
            parsed_object = mean_reward_parser.parse(line)
            if parsed_object is not None:
                return float(parsed_object[0].strip())
            else:
                return None
```

```
In [4]: fig, ax = plt.subplots(figsize=(8, 6))
        plt.ion()

        fig.show()
        fig.canvas.draw()

        mean_rewards = []
        line = True
        while True:
            line = output.readline()
            if not line:
                break
            mean_reward = extract_mean_reward_from_line(line.decode('utf-8'))
            if mean_reward is not None:
                mean_rewards.append(mean_reward)
                ax.clear()
                ax.plot(mean_rewards)
                ax.xaxis.set_major_locator(MaxNLocator(integer=True))
                ax.set_xlabel('Number of Episodes')
                ax.set_ylabel('Mean episode reward')
                ax.set_title('Mean Reward vs Episodes')
                fig.canvas.draw()
```

```
<IPython.core.display.Javascript object>
```


Mean Reward vs Episodes
Number of Episodes
Mean episode reward