A Primer for Deep Learning

Professor Xun Huang †
Aeronautics and Astronautics, College of Engineering, Peking University

August, 2020

Preface

First, students shall prepare the corresponding hardware/software environment by following Sec. 1.

Next, a guiding example can be followed, step by step, in Sec. 2.

Then, some advanced topics can be tried and homework questions are designed for providing possible guidance.

The note is solely prepared by Prof. Huang, and the current version is still in its very primitive form. Any comments through emails are welcome.

† Email address for correspondence: huangxun@pku.edu.cn

1. Preliminary Information

The testing computer environment is Macbook Pro (OSX 10.14) with an Intel Iris Plus GPU 655 (1536 MB). Such a hardware setup shall be well below most of your own computers. Please Google/Baidu and find out alternative solutions in case your own system is Windows/Linux.

I would stick to Mac OSX system, which is very similar to Linux. Windows users please try to solve issues by Googling. However, I do encourage you to start learning Linux from now on, and a good starting point is to use Docker/virtual machine.

About control theory, please take this course to learn what it is.

About Python, a good starting point can be found at:
https://github.com/Mallcock1/2016-01-11_Sheffield_Notes

2. Case I: Pole-Cart Problem

2.1. *Installation*

(i) Install Python 2, Python 3, and pip.
(ii) Anaconda may be a more easy way.
(iii) Activate a virtual environment by 'python3 -m venv plaidml-venv', and 'source plaidml-venv/bin/activate'; or by 'conda create py26'.
(iv) In the virtual environment, install TensorFlow (1.14.0, by: python -m pip install –upgrade https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-1.14.0-py3-none-any.whl, numpy shall be installed separately) and Keras (2.2.4). Hint: Do NOT install the most updated versions because the code packages would be incompatible†
(v) Install plaidml-keras to enable cross-platform hardware (read "plaidml/install.rst ... pdf").
(vi) Setup plaidml by typing 'plaidml-setup'. Hint: Choose your own GPU card option.
(vii) install OpenAI Gym (https://github.com/openai/gym)

2.2. *Test*

Open Python3, go through the code, 'Pole-cart_Xun.py' (about the Gym env 'CartPole-v0').

2.3. *Try the RL agent*

Install keras-rl (read: Using Keras Reinforcement Learning API with OPENAI GYM).

Open Python3, go through the code, 'Pole-cart_Xun2.py' (about the Gym env 'CartPole-v1').

2.4. *Report*

Email me a report by next Wednesday and answer the following questions:
(i) Make the model work.
(ii) What does DQN mean?
(iii) Why the best reward is 500?
(iv) What does 'SARSAAgent' mean?
(v) modify the network size and compare the different performance.
(vi) (Optional): Explain the agent and the policy (EpsGreedyQPolicy).
(vii) (Optional and advanced topic): Explain the whole coding structure inside the agent, such as what is forward, backward, and step and the associated connections?
I will announce the best report and invite him/her to present the work next Friday.

† Two solutions can be tried to resolve the issue: (1) reinstall everything by: python -m pip install Keras==2.3.1 tensorflow==2.1.0; (2) put the following two lines in your code: import tensorflow.compat.v1 as tf, tf.disable_v2_behavior(), where ',' denotes Enter key.

3. Follow up study of the CartPole case

3.1. *Introduction*

In the previous section, you have learned how to start deep learning environment, and what is its possible applications in practical problems. But all the code are highly encapsulated, which means the implementation details are hidden in classes, that would prevent further understanding of the specific concepts. Hence, here I suggest some further readings and tests.

3.2. *Steps*

The understanding can be deepened by following the steps below.

(i) I always use Keras because of its simplicity and cross-platform.

(ii) Go through and understand every line of the code on: https://keras.io/examples/rl/actor_critic_cartpole/.

(iii) A much more complicated code can be found from keras-rl: https://github.com/keras-rl/keras-rl. It consists of many routines and functions and examples.

(iv) First install it and then focus on the dqn example: dqn_cartpole.py.

(v) Debug this code (by 'import pdb'and 'pdb.set_trace() 'and step through the whole code), and write down the calling connections between classes and functions.

(vi) Start to modify the dqn code, dqn.py, which is generic and cumbersome, to finally understand the basic concept.

(vii) Google every question you have about any unknown python grammar and Keras issues.

3.3. *Test*

Here I suggest a new question to testify whether you really understand and know how to solve your own problem:

Q: Modify the cart-pole case. The current one only control the attitude of the pole (to hold at $\theta = 0\,\text{deg}$). Now the new task is to control both the attitude of the pole and the position of the cart, i.e., $\theta = 0$ and $x = 0$.

Explanation: Physically, during the launch of a rocket, it's attitude shall be normal to the ground and its x position is fixed too. In dqn, the new question suggests that a neural network model shall output two Q values for both θ and x.