

Society for Artificial Intelligence and Deep Learning

Winter 2018 Induction Assignment

December, 2018

1 About

This is the official Winter assignment for SAiDL inductions. Please fill this google form for registration [\[Link\]](#) and Join our Slack channel [\[Link\]](#) and at the end of the assignment please fill the google form for submission [\[Link\]](#) . If you have any doubts first try to Google it, if still not clear you can pm any one of us. The deadline for the assignment is **15 January 2019**.

2 Getting Started

2.1 Preparing your Machine

- Dual boot your machine with Ubuntu **16.04** LTS
- Install git
- Install the latest version of Anaconda with Python 3.6 (make sure that you agree to change PATH variable when prompted)
- Install OpenCV via python for pip
- Install latest version of Pytorch with or without GPU support depending on your system specs.

2.2 Setting up your Github account

Set up a github account and make sure that you redeem the Student Developer plan [\[Link\]](#).

3 Study Resources

3.1 Recommended:

3.1.1 Python 3.6

Python is the Franca Lingua of Ai. You will be learning Python 3.6 . Do it either from “Learn Python 3 the Hard way by Zed Shaw” or tutorial series by sentdex [\[Link\]](#).

3.1.2 Andrew Ng's Coursera Course on Machine Learning

This course is the launchpad for most Ai enthusiasts and gives you your first hands on approach to Machine Learning along with the maths behind it. You will be implementing ML algorithms in Octave/MATLAB.

3.1.3 Linux Terminal

This is one of the fundamental requirements for any computer scientist. This [\[Link\]](#) will help you get started.

3.1.4 Numpy

Crudely speaking this is MATLAB for Python. I suggest you do this after the Andrew NG course. You can go to its official tutorial or learn it with hands on deep learning experience via deeplearning.ai's first course.

3.1.5 Pandas

This is one of the most crucial and powerful libraries in data science. To begin with you end to learn how to read and write CSV and JSON files as well how to manipulate Dataframe rows columns and contents. Again sentdex [\[Link\]](#) to the rescue.

3.1.6 Matplotlib

Learn how to plot basic graphs. The pyplot submodule should be enough for the beginning. Sentdex [\[Link\]](#) is your saviour again.

3.1.7 OpenCV

This is one of the best computer vision library out there, However currently we will be using it only to load images as Numpy arrays. Tutorial for the same is [here](#).

3.1.9 Stanford's CS231n

This is the main course of the entire assignment. You have to see all the lecture videos of Stanford's Class for Computer Vision using Deep Learning. Make sure you do the Spring 2017 iteration. If possible do the course assignments too. [\[Youtube link\]](#) [\[Course Link\]](#)

3.1.8 Gym

OpenAI's gym is a toolkit for development of reinforcement learning algorithms. It has a lot of agents and environments for which state, action, observation, returns based on actions, etc can be generated. You can follow the official documentation.

3.1.9 Mujoco-py

Mujoco-py is a library which together with Gym provides real world physics simulations. Obtain a free student license from [\[Link\]](#). Apply for this at the earliest as it can sometimes take time for the license key to arrive. After your license key arrives, carefully follow the steps in the install mujoco-py section here [\[Link\]](#).

3.1.10 UCB's CS294

This course is on Deep Reinforcement Learning. Watching just the first two lectures of this series will suffice for doing the imitation learning question in the assignment. [\[Course Website\]](#)

3.1.11 Pytorch Tutorials

You will be doing the assignments in Pytorch and Numpy only. We would suggest that you get hands on experience with Pytorch by following the official tutorials or this series [\[Link\]](#).

4 Assignments

The assignments are structured such that you can either solve the first four questions **OR** do the question 5 individually. Keep in mind that solving the last question means replicating state of the art results of a paper.

4.1 Numpy neural network

In this question you will create a neural net in **numpy** which either computes the xor or the xnor of two two bit binary numbers depending on the value of a third input. If the third input is 0, the neural net should compute the xor of the two numbers and if the third input is 1, the neural net should compute the xnor of the two numbers.

4.2 Word Embeddings

Download the Large Movie Review Dataset from Stanford [\[Link\]](#). Extract all the sentences out of the dataset. Now make a model which will learn the word embeddings of all the words present in this dataset. There is no limitation as to which model you use for developing the embeddings. You have to make the model in Numpy(suggested) or pytorch. Most commonly used are word2vec, glove and elmo. Keep in mind that you have to write down and train the model. Using an api which directly returns the embedding, does not count as completing the question.

After learning the word embeddings(word vectors) visualise the embeddings using manifold learning algorithms. (Hint:t-SNE)

You can learn more about word embeddings from Stanford's 224n [\[Link\]](#) course or from the skymind.ai word2vec tutorial.

4.3 Image Generation

Train a generative adversarial network to generate images using the CIFAR10 dataset. There is no limitation to the kind of GAN you train. Report two results: One should be a baseline for your second result, and the second should be an optimisation or training technique which helped in stabilising your training of the GAN and trained it better.

4.4 Imitation learning

Implement imitation learning algorithm on mujoco-py's Humanoid agent. This problem is a part of homework 1 of the UCB CS294 course on Deep Reinforcement Learning [\[Link\]](#). Clone their homework repository and run `run_expert.py` file inside homework 1 directory to generate observation action pairs. Train a neural network such that it achieves appreciable returns. The neural network should be written in pytorch.

Now test your neural network by generating actions from your neural network based on observations from the Humanoid world. To learn about the imitation learning algorithm, watch UCB CS294's first two lectures.

5. Paper Implementation

If you're attempting this question your task is to implement a deep learning paper. Select any one of the papers published in NeurIPS 2018. First read the abstract and give a rough reading of the paper you have chosen. **At this point inform us that you're attempting to implement the paper and tell us about the paper chosen.** Once you get a approval from us you can go ahead and give a through reading of the paper and try implementing the paper using pytorch.

We are not expecting you to write a very clean implementation but will be testing you on your understanding of the paper. Your code should be able to replicate results described in the paper.

Remember this task has not been designed for beginners but anyone is welcome to give it a try.

6. Submission

After completing the assignment (4 questions or paper implementation) upload your code on Github in a repository titled SAiDL-Winter-Assignment-2018. The Deadline is 15 January 2018. You can learn how to use git from codeacademy [\[Link\]](#). To submit fill this google form [\[Link\]](#). Inform us when you complete each question from the assignment (submit the google form only after completion of the whole assignment or after deadline).

7. Contact Info

If you have any questions/suggestions or need guidance on anything pm any one of us on slack or mail us. Join the slack group for updates.

Our Team -

Pranav Garg : f20150295@goa.bits-pilani.ac.in (Monk)

Alish Dipani : f20160005@goa.bits-pilani.ac.in

Mehul Rastogi : f20160087@goa.bits-pilani.ac.in

Sharad Chitlangia : f20170472@goa.bits-pilani.ac.in

Rijul Ganguly : f20170971@goa.bits-pilani.ac.in