Society for Artificial Intelligence and Deep Learning

https://sforaidl.github.io/

Winter 2019 Induction Assignment

December, 2019

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 at first try to Google it, if still not clear you can pm any one of us. The deadline for the assignment is **3rd January 2020**.

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].

2.3 General Notes

Before you try and attempt the assignment, it is important that you remember we value your attempt more than the final results. You may or may not have completed the assignment as a whole, but make sure to submit it.

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 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. Probably one of the best *practical* courses out of all the Reinforcement Learning courses due to its assignments. This course may help you if you attempt Question 5. [Course Website]

3.1.11 Pytorch/Tensorflow Tutorials

You will be doing the assignments in Pytorch or Tensorflow 2.0 only. We would suggest that you get hands on experience with Pytorch by following the official tutorials or this series [Link]. If you are using Tensorflow, make sure to use TF 2.0 or greater. If your solutions are in TF < 2.0, they won't be evaluated as they are not valuable.

3.1.12 LaTeX Tutorials

You will be writing your Paper Review in LaTeX. From writing resumes/CVs to writing research papers and even writing equations in Jupyter Notebook, learning LaTeX helps. Follow the tutorial at [Link] to learn LaTeX. Only the first three modules will be required for this assignment. Use overleaf to create your LaTeX documents.

4. Assignment

Depending on your experience and knowledge you can either choose to attempt three different questions(4.1, 4.2, 4.3) or one long research question(5). If you attempt question 4 all three questions are compulsory. If you choose to attempt 5, then you don't need to attempt any of the questions of {4.1, 4.2, 4.3}.

4.1 Optimizer study

In this question you will have to study different optimizers (SGD, Adam, Rectified Adam, Adagrad, RMSProp, etc). Create and train a simple neural network (keep your network complex enough to see the difference in optimizer performance e.g. Resnets) and study the convergence of this network using different optimization methods. Write your optimizers **from scratch** and try to mimic the purpose for which they were created. Bonus points for doing everything in Jax rather than Pytorch. The idea is to study different optimization

methods. Don't limit yourself to just the optimizers given above, you may also explore different optimization methods given in the http://www.deeplearningbook.org/.

Jax is a Python library which augments numpy and basically will make it much easier for you to work with the gradients needed for this question. You can learn more about Jax from here or here.

4.2 Applied Deep Learning

Attempt either A or B.

A. Transformers v/s LSTMs [Hard]

Perform a comprehensive comparative study of the performance difference between Transformers and LSTMs. You may look for datasets here. Select datasets appropriately for the same. Some ideas for the same are given below:

- Do LSTMs perform better with less data than transformers?
- How does attention help over LSTMs given that both are primarily prescribed for keeping the 'focus' on a particular part of the sentence?
- Compare training steps v/s parameters. Bonus for configuring tensorboards for your tests. Also accuracies/losses v/s training steps

By no means is this a comprehensive list, these ideas are just meant to give a nudge in the right direction - *a small gradient step* if you may.

B. Representation Learning [Relatively Easier]

Your task in this question is to study transfer learning for one-shot learning on the popular dataset Omniglot. Build a CNN to perform pre-training on MNIST like datasets. Follow this up with the building of a new novel algorithm/strategy by transfer learning. E.g. Train a CNN on MNIST and use that trained CNN to build a Siamese network to classify if examples are of the same class. This example is not indigenous and is just meant to give you an idea of what is to be done.

4.3 Time Series Research

Time-series classification is a rapidly growing field with applications ranging in a variety of fields from studying brain waves and EEG data (neuroscience), to predicting and correcting information flow in signals (electrical engineering), weather forecasting, power prediction, quantitative finance etc.
In this task, you will need to familiarize yourself with time series data and one of the challenges that are faced while trying to deal with these datasets as a data scientist.

Skewed Classes: Imagine a very basic example of the case of malfunction of a medical instrument device. Consider that the data is designed in a way that you have a 0 if the medical instrument worked correctly (pass), and 1 if the instrument failed to do its job. Obviously, in a real-world scenario, you'd expect to have a lot less 1's than 0's (no of failures

<< no of successes). As such, you'd probably guess, that if you were to build a predictor, it's accuracy (no. of correct guesses) would probably be high if it simply answers 0 all the time. Thus this dataset is what we call a skewed dataset, aka a dataset which has a lot more examples of a few specific classes(here 0) than the other classes(here 1).

Your task is to design a skewed dataset from scratch, and provide an explanation (like the above, but not the same, you can be creative here, just try to provide some rationale) as to why the data would be skewed in the real world. Then you'd need to use any classifier (linear, non-linear, neural, deep, whatever works, impress us if you want) and train it on the dataset. Then, use any popular metric used for evaluating such skewed datasets to correctly estimate the performance of your classifier.
Your points will be based upon originality of thought, and rationale behind the codes based on the supporting comments. Be as descriptive as possible.

You will need to submit a **report in LaTeX** (keep this for the last day) which should contain the description of your dataset, the explanation as to why it will be skewed, a baseline method on your dataset and any method you used to improve the performance of your classifier on the skewed dataset. Follow the IEEE conference report format. Your report does not need to exceed more than 2 pages. Focus on your research work more than beautifying the report.

OR

5. Research Question - Reinforcement Learning in Seq2Seq

As the name suggests, seq2seq models are used to convert input sequences to output sequences while preserving certain representative information. There has been a lot of focus recently on the use of seq2seq models for tasks such as text summarization, text generation and machine translation. The original seq2seq model (Sutskever et al. 2014) used LSTM units to encode sequences into latent meaningful representations which are decoded into a different form using similar architectures. More recently, the LSTM units have been replaced by transformer blocks in a variety of ways including encoder-only (Devlin et al. 2018) and decoder-only (Radford et al. 2019).
Even while performance of these models has greatly improved (especially with BERT and GPT-2), there are few downsides, the most prominent of them being a) exposure bias and b) very high computational requirements.
A new approach to seq2seq problems has been with Reinforcement Learning i.e. by learning useful representations and generation strategies through interaction (Keneshloo et al. 2018).

Your task consists involves exploring, designing and implementing a strategy for an RL-based seq2seq model. You are required to submit 2 things:

a) A report consisting of a literature review of recent approaches and a detailed description of your approach along with results and performance plots. Follow the IEEE conference report format.
b) A GitHub repository containing PyTorch/TensorFlow code for the following:
    i) Baselines that you have run

ii) Code for the approach that you have taken

Keep in mind that we do not expect a completed work but rather look forward to a detailed formulation for a sensible approach and code evidence that suggests that it could be explored further.

6. Submission

After completing the assignment (either 4 or 5) upload your code on Github in a repository titled SAiDL-Winter-Assignment-2019. The Deadline is 3rd January 2020. 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

General doubts or doubts that you think might help others as well should be asked on #general channel. Doubts regarding the first and third question can be asked to anyone. If you have any doubts pertaining to a specific question, then reach out to the person(s) assigned for that task (check details below) first via pm.

Our Team -
Sharad Chitlangia
Rijul Ganguly
Pranav Mahajan
Ajay Subramanian
Souradeep Chakraborty
Rajaswa Patil (Honorable president of the religious Language Research Group)