Project List For IE534/CS547

March 4, 2021

1 Project Requirements

- Each group is suggested to choose one of the projects listed below.
- Some projects may have specific requirements. For the other projects that only have links of the paper, you are required to reproduce the results. Reference to the original code on Github is encouraged. But you should write your own code.
- The files to be submitted include code, a detailed report and slides that take no more than 7 minutes. A presentation may be needed based on the course schedule.
- Extra credit is available for the projects that have any one of the following properties.
  - Modify the existing model structure or propose a new model structure. Prove it has better performance than the model in the paper.
  - Propose new algorithm and prove it has better performance than the algorithm in the paper.
  - Apply the existing model or algorithm in the paper to new datasets.

  The extra credit is up to 7% of the project score. Some other novel ideas are highly encouraged. Groups can discuss the ideas with TAs and the plan will be reviewd by prof. Sowers.
- Projects are proposed by TAs. Some papers may have a lot of results that can hardly be finished in one semester. You are free to discuss the project workload and details with the corresponding TA.

2 List of Projects

TA: Xiaobo Dong

Project 1:Deep Q learning All the papers below are Deep Q Learning rated from the original one to the complicated one. The goal of this project is to explore the DQN reinforcement learning. An ideal path is to go over all the papers from the list and write a brief

summary of each and implement a couple with some well-known environment or your own designed environment.

Playing Atari with Deep Reinforcement Learning, Mnih et al, 2013. Algorithm: DQN.
Link: https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf

Deep Recurrent Q-Learning for Partially Observable MDPs, Hausknecht and Stone, 2015. Algorithm: Deep Recurrent Q-Learning.
Link: https://arxiv.org/abs/1507.06527

Deep Reinforcement Learning with Double Q-learning, Hasselt et al 2015. Algorithm: Double DQN.
Link: https://arxiv.org/abs/1509.06461

Dueling Network Architectures for Deep Reinforcement Learning, Ziyu Wang etc.
Link: https://arxiv.org/pdf/1511.06581 Prioritized Experience Replay,Tom Schaul, John Quan, Ioannis Antonoglou, David Silver
Link: https://arxiv.org/pdf/1511.05952
Rainbow: Combining Improvements in Deep Reinforcement Learning, Hessel et al, 2017.
Link: https://arxiv.org/abs/1710.02298

Project 2:Policy Gradient Reinforcement Learning

All the papers below are Policy Gradient reinforcement learning rated. The goal of this project is to explore the Policy Gradient reinforcement learning. You can start from the original Policy Gradient method and Actor-Critic Method. An ideal path is to go over all the papers from the list and write a brief summary of each and implement a couple with some well-known environment or your own designed environment.

Trust Region Policy Optimization, Schulman et al, 2015. Algorithm: TRPO.
Link: https://arxiv.org/abs/1502.05477

Proximal Policy Optimization Algorithms, Schulman et al, 2017
Link: https://arxiv.org/abs/1707.06347

Project 3:Day Trading

This is an open but very risky project. The goal of project is to explore the ability of day trading with Machine learning method. The data could be achieved from Yahoo Finance. In order to design your own trading system, you need to select your own trading signals, like moving average, or some content embedding of finance news(NLP). Moreover, a trading model, like reinforcement learning based or other method should be designed. This project could be involved with Reinforcement Learning, Natural Language Processing, Time-series Prediction, and etc. If you would like to choose this one, please Email me directly with your own detailed idea.

Project 4:Meta-Learning

All the papers below are Meta-Learning rated. The goal of this project is to explore Optimization-based and Metric-based Meta-Learning.

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, Finn et al, 2017. Algorithm: MAML.
Link: https://arxiv.org/abs/1703.03400

On First-Order Meta-Learning Algorithms Alex Nichol, Joshua Achiam, John Schulman.
Link: https://arxiv.org/pdf/1803.02999.pdf

Matching Networks for One Shot Learning.
Link: http://papers.nips.cc/paper/6385-matching-networks-for-one-shot-learning.pdf

Prototypical Networks for Few-shot Learning
Link: https://arxiv.org/pdf/1703.05175.pdf

TA: Lei Fan

Project 5: Convolutional Neural Networks for Text Classification.

Implement the following two papers:

- Very Deep Convolutional Networks for Text Classification
  Link: https://arxiv.org/pdf/1606.01781.pdf
- Densely Connected Convolutional Networks
  Link: https://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.pdf

Requirements:

- Implement the models in the above two papers and test on the following two dataset: Yahoo answers dataset and yelp review polarity dataset (links are below).
- You can use a smaller number of layers for the densely connected CNN.
- Feel free to train and test only on some part of the dataset if the whole dataset is too large. But please clearly mention which part of data you use in each of the dataset.

Data:
https://www.kaggle.com/soumikrakshit/yahoo-answers-dataset
https://www.kaggle.com/irustandi/yelp-review-polarity/version/1

Project 6: Attention and Transformer

Attention Is All You Need.
Link: https://arxiv.org/pdf/1706.03762.pdf
This project is about attention and the transformer. They are very popular models in NLP. Coding the whole structures by yourself is highly encouraged.

Project 7: BERT

Bert: Pre-training of deep bidirectional transformers for language understanding. J Devlin, MW Chang, K Lee, K Toutanova

Link: https://arxiv.org/pdf/1810.04805.pdf&usg=ALkJrhhzxlCL6yTht2BRmH9atgvKFxHsxQ

Project 8: Neural Hawkes

Implement the Neural Hawkes model for sequential events prediction. Reproduce the results on stack overflow and the retweet dataset.

Description:

- We focus on the problem that: given a series of arrival time and types of action, we predict the next event time and type. Arrival time are generally modeled using point process models. So this project is mathematical.
- Neural Hawkes replace the intensity function of the Hawkes process with a continuous-time LSTM.
- Since we have relatively small public datasets, training doesn't take too much time. CPU is enough.
- In this project, you need to use mask to deal with sequences with various length.
- Grading is based on: your accuracy on action types prediction and rmse for arrival time prediction.

Code: https://github.com/HMEIatJHU/neurawkes
Paper: https://arxiv.org/pdf/1612.09328.pdf
Data: https://drive.google.com/drive/u/0/folders/12RLDZTfsR6OF2pFJA10d6YSPFDu5YDRL

TA: Rachneet Kaur

Project 9: Image to image translation using cycle GANs

Implement a Cycle-consistent Generative Adversarial Network (CycleGAN) model from scratch for image-to-image translation problems based on the paper [1].

This model can translate an input image to another image based on the model parameters. If the model parameters are trained on image collections of horses and zebras, then the model can translate an input horse image into an output zebra image in terms of color. If the model is trained on image collections of photos and art paintings, then the model can output an image in the art style of the input image. This technique is useful for paired images generation including collection style transfer, object transfiguration, season transfer, photo enhancement.

The datasets for the implementation are available at [2].

Paper: https://openaccess.thecvf.com/content_ICCV_2017/papers/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.pdf
Data: https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/
Code: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix
Another Helpful Link: https://cyclegans.github.io/

Expectations:

Successful implementation on at least one of the datasets.

- Visual results similar as in https://cyclegans.github.io/project1/2018/04/29/Results-and-Analysis-on-Cycle-GAN-Implementation/
- Loss function values similar to Tables 1-5 in paper [1].

References:

1. Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks." In Proceedings of the IEEE international conference on computer vision, pp. 2223-2232. 2017.
2. https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/

Project 10: Show and Tell - A Neural Image Caption Generator

Build a generative model using latest techniques in Computer Vision and Machine Translation to describe an image. This project involves implementation of the paper [3]. In this project, a single joint model is constructed to generate a target sequence of words that describes the input image. The idea is similar to encoder-decoder RNN used in translating sentences except that a Convolutional Neural Network is used in place of the encoder RNN. CNN, which takes image as an input, is trained for an image classification task to generate a compact representation of the original image. This representation is then passed as an input to a decoder RNN that generates the sentences.

Paper: https://arxiv.org/abs/1411.4555
Data: MSCOCO 2014 [4] and Flickr30k [5]
Code: https://github.com/kelvinxu/arctic-captions

Expectations:

Implementation on at least one of the datasets and performance evaluation using BLEU, CIDEr and ROUGE_L metrics similar to as follows.

Days	BLEU1	BLEU2	BLEU3	BLEU4	CIDEr	ROUGE_L
COCO	64.6 | 71.3	45.9 | 54.2	31.7 | 40.7	22.0 | 27.7	69.4 | 85.5	47.6 | 53.0
Flickr30k	53.9 | 63.0	34.8 | NA	22.0 | NA	14.3 | NA	26.5 | NA	39.5 | NA

Table 1: There are two results in each cell. The left is for our code. The right is for the results in the paper.

References:

3. Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. "Show and tell: A neural image caption generator." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3156-3164. 2015.

4. Chen, Xinlei, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO captions: Data collection and evaluation server." arXiv preprint arXiv:1504.00325 (2015).

5. Plummer, Bryan A., Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to sentence models." In Proceedings of the IEEE international conference on computer vision, pp. 2641-2649. 2015.

Project 11:

Faster RCNN Implement "Faster R-CNN: Towards real-time object detection with regional proposal networks", NIPS, 2015 by Ren et al.
This is a challenging project.
Link: https://arxiv.org/pdf/1506.01497.pdf

Project 12: Image Ranking

Implement a deep learning model for image ranking. The goal of this project is to introduce you to the computer vision task of image similarity. Like most tasks in this field, it's been aided by the ability of deep networks to extract image features. The task of image similarity is retrieve a set of n images closest to the query image. One application of this task could involve visual search engine where we provide a query image and want to find an image closest that image in the database. Your task, for this project, will be to implement a simplified version of the pipeline introduced in "Learning Fine-grained Image Similarity with Deep Ranking".
We strongly encourage you to read this pager - https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/42945.pdf
Here's another good resource - https://medium.com/@akarshzingade/image-similarity-usingdeepranking-c1bd83855978
Further details on https://drive.google.com/file/d/1F_GeIO5ADe95FSSBk4suwwHhRgG1tymz/view?usp=sharing