Due: 2019/12/30

Course Project

Computer Science Department in SJTU

1 Brief Introduction

1. 3 - 4 people for each group.
2. This project is group work. Each group is required to choose at least two topics from the given four topics as your course project and submit a report by the end of this term. Topic D is encouraged by 10% bonus.
3. Finally, each group should submit source code and report, accounting for 30% and 70% of the total score respectively.
4. **Submission deadline: 8:00 pm, 2019/12/30**
   ftp://public.sjtu.edu.cn (username: huangwenjing password: public)

2 Description for topics

2.1 Topic A

How does the sample size influence the smoothness of the loss function (for example, the number of local optimal solutions and the difficulty of jumping out of the local optimum)? Conduct experiments to support your claims and explain why.

2.2 Topic B

How does the depth of network influence the smoothness of the loss function (for example, the number of local optimal solutions and the difficulty of jumping out of the local optimum)? Conduct experiments to support your claims and explain why.

2.3 Topic C

How does the number of iterations influence the smoothness of the loss function (for example, the number of local optimal solutions and the difficulty of jumping out of the local optimum)? Conduct experiments to support your claims and explain why.

2.4 Topic D

Will bi-directional learning be helpful to overcome the problem of getting stuck in the local optimum? (refer to [4, 5]) Conduct experiments to support your claims and explain why.

2.5 Topic E

Investigate the influence of gating mechanism to the skip connections in the neural network (for examples, ResNet [3]) for the same task. Conduct experiments to support your claims and explain why.

2.6 Topic F

Compare the difference of using pixel leave error (such as mean square error) and discriminative loss (computed by the discriminator in GAN [2]) for the same image reconstruction task (for example, image inpainting).

2.7 Topic G

Choose one task (for examples, AlphaGo) and implement A* algorithm [6] and Monte Carlo tree search (MCTS) [1]. Compare the difference between A* algorithm and MCTS. Further, propose a developed search algorithm by combining the advantages of A* and MCTS.

3 Basic Assignments

1. Choose an optimization problem (for example, classification, reconstruction, etc.) and train the models to optimize the problem.
2. Evaluate the performance of your trained models and investigate the smoothness of loss functions.
3. Write a report that contains your ideas, methods, algorithms, experimental details and results.

3.1 Codes

Each group is required to submit your codes by providing a link to your GitHub repo (attached at the end of your report). If you do not know how to use GitHub, please visit its tutorial[1] for some advice. Codes will be judged by cleanness and readability, so remember to comment on your codes.

3.2 Report

Each group is required to turn in a report that contains your main ideas, methods, algorithms, experimental settings, and results. You can write your report in word or latex (encouraged), English (encouraged) or Chinese. The details of the four parts are:

[1] https://guides.github.com/activities/hello-world/

Main Ideas A brief introduction of your report, including your proposed methods and the performance.

Methods The proposed method for the project, including a detailed description of your methods.

Algorithms The algorithms you choose to optimize your model (how to train your model or update the parameters), including the motivation, detailed description and pseudo-code if necessary.

Experimental Settings The experimental settings, including the structure of your network (if you have), learning rate, batch size, etc.

Results The performance of your model on the test set.

References

[1] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[5] Lei Xu. Least mean square error reconstruction principle for self-organizing neural-nets. *Neural networks*, 6(5):627–648, 1993.

[6] Lei Xu. Deep bidirectional intelligence: Alphazero, deep ia-search, deep ia-infer, and tpc causal learning. *Applied Informatics*, 5(1):5, Sep 2018.