

CS280 2018 Fall Midterm Project
Striving for Simplicity: the All Convolutional Net

Due date: Nov 20 , 2018

Overview

In midterm project, you are required to implement All Convolutional Network in [paper](#) and reproduce experiments results on CIFAR-10 datasets. Then you should finish transfer learning on CIFAR-100 dataset. Details of todo list is described in Sec.1 and Sec.2. **Pytorch is the only deep learning framework you can use in this project.** You will be evaluated based upon:

- Paper understanding
- Data preparation
- Modeling
- Evaluation and reproduced results
- Code quality
- Writing

Output

You are required to submit the following things to git classroom, [here is the link](#).

- All your source code(except dataset)
- Midterm project report(CVPR2019 format)

In your report, please include at least

- Experiment environment, including operating system version, pytorch version.
- The details of your experiment setting, including model revision, weight initialization strategy, hyper-parameters, data augmentation and preparation, dataset split etc.
- Experiments results with necessary tables, chart and analysis. You should append corresponding screenshot of your evaluation and training log in appendix at the same time. **You will get 0 points for this part without screenshot.**
- Appendix. Appendix 1 includes code structure screenshot and short explanation for each file. Appendix 2 includes screenshot of your evaluation and training log. Appendix 3 includes loss, accuracy and other curves for every experiment you show in report.
- Citation of any resource/code/reference you use.

1.All-Conv Net on CIFAR-10 dataset

- Implement three base networks A, B, C used for classification on CIFAR-10. Architectures of these three models are described in Table 1 in paper
- For base network C , implement its three additional variants respectively. For details of derived models, please refer to Table2 in paper
- Reproduce the results of base networks A, B, C and the three variant networks of network C on the CIFAR-10 dataset. You can refer the official results in table3.

2. Transfer learning on CIFAR-100 dataset

After you train your model from scratch on CIFAR-10, we want you to practice transfer learning on another dataset, CIFAR-100. There are quite a few options to do transfer learning and we refer you to the note from CS231n (<http://cs231n.github.io/transfer-learning/>).

- Try to use model 'ALL-CNN-C' in table2 trained on CIFAR-10 as a feature extractor. Replace and retrain the final FC classifier, while fine-tuning the parameters of other layers. Compare your results with training the whole model from scratch. (You need to do experiments on two given subset of CIFAR100)
- Analysis the results on two given subset of CIFAR100 in you reports.

In the report, you should include the details of your experiment settings and results, including model revision, weight initialization strategy, hyper-parameters, etc. You can show your design and analysis by tables or charts.

3. Bonus(optional)

1. Reproduce the results in table3, ALL-CNN with data augmentation on CIFAR10. (The paper can get 7.25% error.)
2. Explore the other transfer strategy on two given subset of CIFAR100 and reach your conclusion in report.

4. Two Subset of CIFAR100

We pick up two subsets of CIFAR100, which are *class1* and *class2* with almost the same format as CIFAR10. Each subset contains 10 classes, where *class1* has the similar categories as CIFAR10 while *class2* has much different categories. You can check the categories in variable *fine_label_names*. You can download the subsets from [Google driver](#).