Homework 2: Convolutional Neural Networks and Beyond

Deep Learning (84100342-0)
Spring 2019
Tsinghua University

1 Introduction

As we have learned in class, CNNs are made up of layers with learnable parameters including weights and bias. Each layer takes the output of previous layer to perform some operations and produces an output. Recent years, main-stream CNNs, such as AlexNet [3], VGG [5], GoogleNet [6], ResNet [1], DenseNet [2] and so on, have achieved increasingly better performance on ImageNet dataset and have been leveraged to a diverse applications across many fields.

In this homework, a dataset of clip-art style is provided and you are required to solve a problem of image classification by convolutional neural networks (CNNs). You need to finish the following three tasks on the given dataset:

- **Task A**: Download a ResNet-50 model pretrained by ImageNet, finetune it on our dataset and thus get a new model named as **A**.
- **Task B**: Use the architecture of ResNet-50 and train a new model named as **B** from scratch.
- **Task C**: Design and implement your own convolutional neural networks (CNNs) and train a new model named as **C** from scratch.

2 Dataset

- The dataset contains 65 categories of clip-art images.
- Directory structure:
  - **training set**: "`./train/{0,1,...,64}/xxx.jpg`"
  - **validation set**: "`./valid/{0,1,...,64}/xxx.jpg`"
  - **test set**: "`./test/xxx.jpg`"

Please note that the test set will be released at 5:00 pm on the last day (7 hours before deadline). So please prepare well to test your specified model **C** (**Note: only model C is required**) as soon as the test dataset is available and submit your prediction results in time. To evaluate the performance of your model, you are required to submit a `.txt` file using the following format. It contains the predicted label for every image in the test set. There are two items separated by a comma in each row where the first is the image identifier and the second represents its predicted label.

```
id,label
0001, 2
0002, 3
0003, 3
0004, 3
......
```

3 Requirements and Evaluations

3.1 Programming Language

Python only.

3.2 Deep Learning Framework

We recommend PyTorch and TensorFlow. If using other frameworks, please contact TA.

3.3 Tutorials

- **pytorch**
  - `https://pytorch.org/tutorials/`
  - `https://github.com/utkuozbulak/pytorch-cnn-visualizations`
- **tensorflow**
  - `https://www.tensorflow.org/tutorials`
  - `https://github.com/tensorflow/lucid`
- **Lab 1 and Lab 2 of this course**.

3.4 Scoring

- **Train and evaluate your models**:
  - Train a model **A** for Task A and evaluate its accuracy on validation set [**7.5pts**].
  - Train a model **B** for Task B and evaluate its accuracy on validation set [**7.5pts**].
  - Design and train a model **C** for Task C and evaluate its accuracy on validation set [**20pts**].
  - Besides only evaluating the accuracy of each model, **plot the confusion matrix** of these three models on validation set [**10pts**].
- **Babysit your model C**:
  - Use the techniques of **weight Initialization**, **data augmentation**, **learning rate strategy** you have learned in class to improve the performance of your model **C** from scratch and give a detailed ablation study in your report [**10pts**].
  - Compare the performance of your model **C** under different conditions: [**10pts**].
    - SGD or Adam
    - *max-pooling* or *average-pooling*
    - w/ or w/o dropout
    - w/ or w/o Batch Normalization
  - Use extra techniques you find in other materials to further improve your model. Please explain why you choose it and how it works [**5pts**].
- **Dig into your model C**:
  - Visualize features of the last *fc* layer using t-SNE [4] [**10pts**].
  - Leverage neural network visualization toolkits to visualize some *conv* features [**10pts**].
    - **tensorflow**: `https://github.com/tensorflow/lucid`
    - **pytorch**: `https://github.com/utkuozbulak/pytorch-cnn-visualizations`
    - **others**: Any other toolkits if you think they are helpful...
- **Evaluate your model C on test set** [**10pts**].
  - You just need to submit a `.txt` file containing the prediction results of your model C.
  - TAs will evaluate your prediction results and give you a feedback.

3.5 Notification

- Please submit your code, document and trained models (**A, B, C**) as an Archive (zip or tar). The document is supposed to cover your **insights** of the proposed model, the **technical details**, the **experimental results** (including training and validation curves), and the necessary **references**. Please **DO NOT** use additional data sources.
- We will focus on your code and document to decide your score. Your proposed model needs to be different from the main-stream ones, such as AlexNet [3], VGG [5], GoogleNet [6], ResNet [1], DenseNet [2] and so on. Still, under equal conditions (novelty, code quality, document quality), a higher accuracy along with reasonable computation efficiency contributes a higher score.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[2] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[4] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9, 2015.