EE 569: Homework #6 Successive Subspace Learning

Issued: 04/08/2020
Problem 1&2 due: 11:59PM, 04/26/2020
Problem 3 due: 11:59PM, 05/03/2020

General Instructions:

1. Read *Homework Guidelines* for the information about homework programming, write-up and submission. If you make any assumptions about a problem, please clearly state them in your report.
2. You need to understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.
3. You are required to use *Python* for this Homework.

Problem 1: Understanding Successive Subspace Learning (SSL) (50%)

(a) Feedforward-designed Convolutional Neural Networks (FF-CNNs) (20%)

When two CNN layers are cascaded, a non-linear activation function is used in between. As an alternative to the non-linear activation, Kuo *et al.* proposed the Saak (subspace approximation via augmented kernels) transform [1] and the Saab (subspace approximation via adjusted bias) transform [2]. Specifically, Kuo *et al.* [2] proposed the first example of a feedforward-designed CNN (FF-CNN), where all model parameters are determined in a feedforward manner without backpropagation. It has two main cascaded modules:
1) Convolutional layers via multi-stage Saab transforms;
2) Fully-connected (FC) layers via multi-stage linear least squared regression (LLSR).
Although the term "successive subspace learning" (SSL) was not used in [2] explicitly, it does provide the first SSL design example.

Read paper [2] carefully and answer the following questions:

(1) Summarize the Saab transform with a flow diagram. Explain it in your own words. The codes for the Saab transform can be found at https://github.com/USC-MCL/EE569_2020Spring.
Please read the codes with the paper to understand the Saab transform better.

(2) Explain similarities and differences between FF-CNN and backpropagation-designed CNN (BP-CNNs).

Do not copy any sentence from any paper directly, which is plagiarism. Your scores will depend on the degree of your understanding.

(b) Successive Subspace Learning (SSL) (30%)

Two interpretable models adopting the SSL principle for the image classification task were proposed by Chen *et al*. They are known as PixelHop [3] and PixelHop++ [4]. Read the two papers carefully and answer the questions below. You can use various tools in your explanation such as flow charts, figures, formulas etc. You should demonstrate your understanding through your answer.

(1) Explain the SSL methodology in your own words. Compare Deep Learning (DL) and SSL.

(2) What are the functions of Modules 1, 2 and 3, respectively, in the SSL framework?

(3) Explain the neighborhood construction and subspace approximation steps in the PixelHop unit and the PixelHop++ unit and make a comparison. Specifically, explain the differences between the basic Saab transform and the channel-wise (c/w) Saab transform.
(4) Both PixelHop and PixelHop++ use the Label-Assisted reGression (LAG) unit for supervised feature dimension reduction. Explain the role and the procedure of LAG in your own words. What is its advantage?

Problem 2: CIFAR-10 Classification using SSL (50%)

(a) Building a PixelHop++ Model (35%)

An example block diagram of the PixelHop++ framework containing three PixelHop++ Units is shown in Figure 1. The codes for the c/w Saab transform module, feature selection (cross-entropy computing) module and the LAG unit are provided in the GitHub link. You can import them in your program to build your model in Python based on the diagram. Note that in the first PixelHop++ unit, RGB three channels are processed together instead of channel-wise processing because they are not decorrelated. For your experiments in this Part (a), follow the parameters indicated in Table 1.

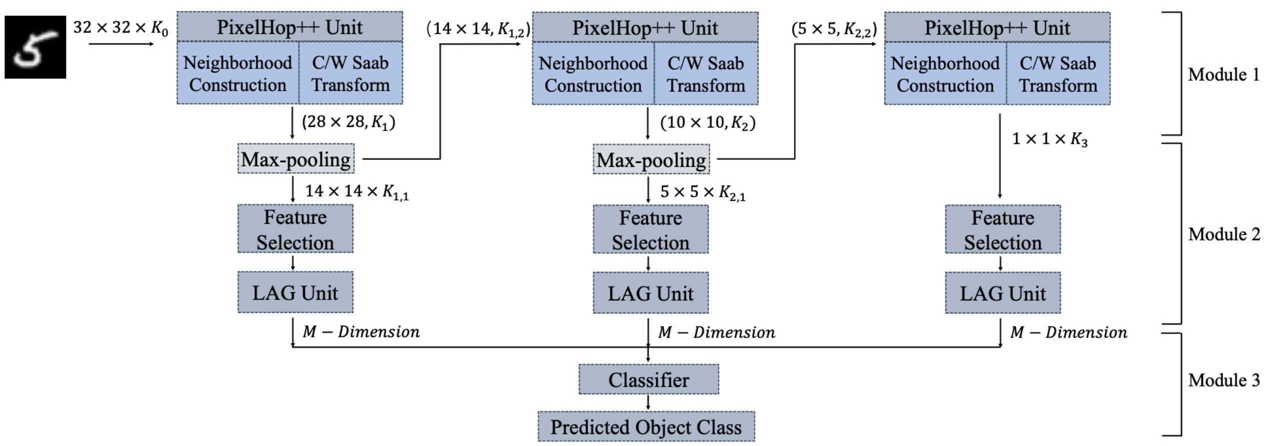

Figure 1 Block diagram of the PixelHop++ model [4]

Table 1 Choice of hyper-parameters of PixelHop++ model for this section

Spatial Neighborhood size in all PixelHop++ units	5x5
Stride	1
Max-pooling	(2x2) -to- (1x1)
Energy threshold for intermediate nodes (*TH1*)	0.001
Energy threshold for discarded nodes (*TH2*)	0.0001
Number of selected features (N_S) for each Hop	Top 50%
α in LAG units	10
Number of centroids per class in LAG units	5
Classifier	Random Forest (recommended)

(1) In order to run locally without memory error, train the Module 1 (PixelHop++ units) with 10,000 randomly selected training images which will take at most 16G of the memory. You need to keep balance among different classes (i.e. randomly select 1000 images per class). Then train Module

2 and Module 3 on all the 50,000 training images. Report training time and train accuracy. What is your model size in terms of the total parameter numbers?

(2) Apply your model to 10,000 testing images and report test accuracy.

(3) Keeping the trained Module 1 unchanged, reduce the number of labeled training samples in Module 2&3 to 1/4, 1/8, 1/16 and 1/32 of the original training number (i.e. 50,000). Apply each model to 10,000 testing images and draw the curve of test accuracy for each setting with respect to the number of training images. Show and discuss your results.

(b) Error analysis (15%)

Most often, a dataset contains both easy and difficult classes. Conduct the following error analysis based on your trained model using 50,000 training images (in Module 2&3):

(1) Compute the confusion matrix and show it as a heat map in your report. Which object class yields the lowest error rate? Which object class is the most difficult one?

(2) Find out the confusing class groups and discuss why they are easily confused with each other. You can use some exemplary images to support your statement.

(3) Propose ideas to improve the accuracy of the difficult classes for PixelHop++ and justify your proposal. There is no need to implement your ideas.

Problem 3: EE569 Competition --- CIFAR-10 Classification using SSL (50%)

The parameters in Table 1 may not be optimal. You are free to modify PixelHop++ in Problem 2(a) to improve its performance in terms of classification accuracy, running speed and model size. For example, you may adjust the hyper-parameters such as the spatial neighborhood size, energy threshold, the reduced dimension through LAG units, etc. You may also increase the number of cascaded PixelHop++ units, adjust the choice of pooling layers (max-pooling, mean-pooling, min-pooling, etc.), and/or other color representations of input images. Besides, you may consider ensembles as done in [5].

Conduct experiments and report your findings in the following aspects.

- Motivation and logics behind your design (20%):
  Draw the diagram of your proposed SSL system and explain it in your own words. Discuss how you choose the hyper-parameters. Discuss the sources of performance improvement as compared with that in Problem 2(a).
- Classification accuracy with full and weak supervision (10%):
  Report the classification accuracy under different training image numbers used in Module 2&3. Draw the test accuracy curve with training models obtained by different training image numbers. Compare it with the curve you obtained using BP-CNN in HW5 Problem 2 with discussion. You can also study the influence of training image numbers used in Module 1 (optional).
- Running time (10%)
  Report the training time and inference time.
- Model size (10%):
  Compute and report the size (i.e. the number of parameter numbers) in your model. It is related to energy threshold (*TH1 & TH2*) and the number of selected cross-entropy-guided features (N_S). Compare it with the model size of your BP-CNN in Problem 2 of HW5.

Your grading will be based on the accuracy, running time and model size of your proposed SSL system in comparison with those of other students in this class.

References

[1] C.-C. Jay Kuo and Yueru Chen, "On data-driven Saak transform," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 237–246, 2018.
[2] C-C Jay Kuo, Min Zhang, Siyang Li, Jiali Duan, and Yueru Chen, "Interpretable convolutional neural networks via feedforward design," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 346–359, 2019.
[3] Yueru Chen and C-C Jay Kuo, "Pixelhop: A successive subspace learning (ssl) method for object recognition," *Journal of Visual Communication and Image Representation*, p. 102749, 2020.
[4] Yueru Chen, Mozhdeh Rouhsedaghat, Suya You, Raghuveer Rao, C.-C. Jay Kuo, "PixelHop++: A Small Successive-Subspace-Learning-Based (SSL-based) Model for Image Classification," *https://arxiv.org/abs/2002.03141*, 2020
[5] Yueru Chen, Yijing Yang, Wei Wang, C.-C. Jay Kuo, "Ensembles of Feedforward-designed Convolutional Neural Networks", in *International Conference on Image Processing*, 2019