CSCI 8000 HW 2

Fall 2024 Special Topics in Efficient Deep Learning Due 11/19/2024

Homework 2: Network Pruning

***GPUs may be needed for speeding up the neural network training process in this homework.**

Description

In this homework you will practice how to prune and fine-tune/masked retrain a Convolutional Neural Network. You will start by loading a pretrained VGG13 model, which is a modified version for CIFAR10 dataset. Then, apply pruning methods (OMP and IMP) to get pruned models. You need to achieve the required pruning sparsity while trying to minimize the accuracy degradation compared to the original pretrained dense model.

Instruction

- The dataset used in this homework is **CIFAR-10**. You may need these packages: PyTorch, NumPy, PyYaml (for reading .yaml file), Matplotlib. The commonly used classifier is Softmax.

- Requirements:
  1. Complete the following functions:
     a. read_prune_ratios_from_yaml()
     b. unstructured_prune()
     c. filter_prune()
     d. apply_pruning()
     e. test_sparity()
     f. masked_retrain()
     g. oneshot_magnitude_prune()
     h. iterative_magnitude_prune()
     i. prune_channels_after_filter_prune()
     j. main()
     k. And any other functions if necessary.

     * *You may customize what function arguments that it needs to take*

  2. Define your layer-wise pruning ratio in a .yaml file. You can find an example in ./vgg13_example.yaml. Before you define layer-wise ratio, you need to find the correct CONV layers' name in the vgg13 model. You need to achieve **80%** overall pruning ratio for unstructured pruning and **40%** overall pruning ratio for filter pruning.
     You may try to play with the layer-wise prune ratio to achieve better final accuracy.

3. Conduct **One-shot Magnitude Pruning (OMP)** and **Iterative Magnitude Pruning (IMP)** using **Unstructured Pruning (80% sparsity)** and **Filter Pruning (40% Sparsity)**, respectively.
   So, you need to obtain 4 pruned models:
   a. Omp_unstructured_0.80_acc_xxx.pt
   b. Imp_unstructured_0.80_acc_xxx.pt
   c. Omp_filter_0.40_acc_xxx.pt
   d. Imp_filter_0.40_acc_xxx.pt

   ** Note: 0.80/0.40 is the model sparsity, and acc_xxx is the final accuracy after retraining. IMP method needs to use at least 3 iterations. Name your models in the above format.*

4. Based on the four pruned models (Requirement 3), plot the following figures:
   a. Draw **one** figure to show the sparsity and accuracy comparison of the four pruned models. X-axis: overall sparsity ratio, Y-axis: accuracy.
   b. Draw **four** figures to show the sparsity mask of one layer.
      i. Arbitrary layer is fine.
      ii. One figure for one model.
      iii. Reshape the 4-D weights to 2-D format (each row represents all the weights from the same filter, number of columns = # of channels * kernal_hight * kernal_width)
      iv. Use one color to represent non-zero weights, use other color to represent zero/pruned weights.

5. Answer the questions listed in prune_channels_after_filter_prune() function.

6. Due to the considerable training costs, you do **NOT** need to repeat the process with 3 different random seeds in this HW.

7. Try your best to improve the accuracy of the pruned model. You may play with the layer-wise pruning ratios, fine-tuning tricks, e.g., lr, schedular, optimizer, augmentations …
   You may check this paper for training tricks:
   Bag of Tricks for Image Classification with Convolutional Neural Networks

Submission

- You need to submit a **zip** file including:
  1. Your code files, **ONLY** main.py, vgg_cifar.py, and YAML files.
  2. Do **NOT** submit original dense model and pruned model with your code to eLC. **Upload** your pruned model on Google drive or OneDrive and put the link in your report.

3. **A report (.pdf) that includes**:
   a. Screenshots:
      i. Testing accuracy of original dense vgg13 model.
      ii. Results of **testing accuracy** and **sparsity (test_sparsity function)** of your **four** pruned model.
      iii. Your .yaml file.
   b. Five (1+4) figures mentioned in Requirement 4.
   c. Answers of the questions listed in prune_channels_after_filter_prune() function.
   d. Link of your models (Google Drive or OneDrive).
   e. Your code. (text or screenshot)

- The **zip** file should be named using the following convention:
  <Last-Name>_<First-Name>_HW2.zip
  Ex: Potter_Harry_HW2.zip

- Note:
  Do not put any print function other than showing the results.
  Comment your code.
  Do not include dataset and models in your submission!

- Upload the **zip (not rar)** file to the eLC before **11:59PM (EST Time) 11/19/2024**.
- Late fee: -10 points/day