Project 1

CSI 4140/5140 Deep learning and applications

Due by 11/07/2024 11:59:59pm

Summary

Develop and train a deep neural network (DNN) using PyTorch to classify images from the CIFAR-10 dataset. Perform a comprehensive ablation study to assess the impact of various methods and hyperparameters on model performance. To deepen your understanding of neural networks, you'll have to manually implement it without using PyTorch's built-in functions.

This project should be completed in groups of 1-2 people. I will not assign group members and you need to find your group member by yourself.
Sign up your group in the Google sheet: Project 1 - Group. You still need to sign up even if you do it individually.

You can use Google Colab or Google Cloud Platform to train the model. Google Colab supports GPU. To use GPU to train the model, refer to the tutorials in the Pytorch official website, like: https://pytorch.org/tutorials/beginner/pytorch_with_examples.html
You can apply for free credits from Google Cloud Platform: https://cloud.google.com/free/?hl=en
I have applied for the free credits from both AWS and Google Cloud Platform as an educator, waiting for their approval. When I get approval, I will let you know.

Description

1. Model Architecture: Your model must include the following layers:

1) At least one fully-connected layer
2) At least one convolutional layer
3) At least one ReLU activation layer
4) Softmax layer (for output classification)

2. **Performance requirement**: The final test accuracy of your model should exceed 95% on the CIFAR-10 dataset. The higher, the better. ResNet VGG DenseNet

3. **Regularization Techniques**: The following regularization methods should be implemented:

1) L2 regularization
2) Dropout
3) At least two data augmentation methods (e.g., random cropping, flipping, etc.)

The above techniques should be implemented and evaluated in the ablation study, but not necessarily used in your model training for getting a good test accuracy.

4. **Optimization Techniques:** The following optimization methods should be implemented:

1) Data normalization
2) Mini-batch gradient descent
3) Gradient descent with momentum
4) RMSprop
5) Adam
6) Cosine learning rate decay
7) Two additional learning rate decay algorithms

The above techniques should be implemented and evaluated in the ablation study, but not necessarily used in your model training for getting a good test accuracy.

5. **Evaluation Metrics & Visualizations**: During the training process, record and visualize the following metrics:

1) Training accuracy over each epoch
2) Test accuracy over each epoch
3) Cost over each iteration

6. **Ablation Study**: Conduct a thorough ablation study to analyze the effects of different methods and hyperparameters on model performance. You are required to study and report the effects on test accuracy, and on one of the following metrics, including convergence speed, training accuracy, cost values, etc. Findings or insights should be discussed. Virtualization is preferred, like figures of accuracy over epoch. Convergence speed refers to the rate at which a model's training process reduces the loss function and approaches an optimal or acceptable performance on the training data.

Ablation study is a crucial element in research papers, providing valuable insights into the contribution of each component of a model or system. About how to conduct ablation study, refer to the Section of Ablation Study in the following papers. You can refer to more papers published in the top AI conferences, e.g., NeurIPS, ICML, ICLR, CVPR, ICCV, ECCV, AISTATS, etc.

- Hrank: Filter pruning using high-rank feature map: https://openaccess.thecvf.com/content_CVPR_2020/papers/Lin_HRank_Filter_Pruning_Using_High-Rank_Feature_Map_CVPR_2020_paper.pdf
- Contrastive representation distillation: https://arxiv.org/pdf/1910.10699

Your ablation study should include, at a minimum, the following aspects (the more aspects, the better):

1) Effect of at least three different learning rate decay algorithms
2) Effect of the initial learning rate
3) Effect of the number of training epochs
4) Effect of different regularization methods, at least including L2 regularization and Dropout

5) Effect of lambda of L2 regularization
6) Effect of different data augmentation methods
7) Effect of different optimization algorithms, at least including gradient descent with momentum, RMSprop, and Adam
8) Effect of Adam' hyperparameter beta1
9) Effect of Adam' hyperparameter beta2

Important Notes:

You are required to manually implement all the required layers and algorithms without using PyTorch's built-in functions. I will review your code, and may use a script to verify that no built-in functions are used. If any prohibited built-in functions are found, you will lose all points associated with those implementations.

For more information about the built-in functions, refer to: Pytorch built-in functions

The following built-in data loader function, data augmentation functions, pooling layers, and normalization layers are allowed to use:

- torch.utils.data.DataLoader
- torchvision.transforms.RandomCrop
- torchvision.transforms.RandomHorizontalFlip
- torchvision.transforms.RandomVerticalFlip
- torchvision.transforms.ColorJitter
- torchvision.transforms.RandomResizedCrop
- Pooling Layers
  - torch.nn.functional.max_pool1d
  - torch.nn.functional.max_pool2d
  - torch.nn.functional.max_pool3d
  - torch.nn.functional.avg_pool1d
  - torch.nn.functional.avg_pool2d
  - torch.nn.functional.avg_pool3d
  - torch.nn.functional.adaptive_max_pool2d
  - torch.nn.functional.adaptive_avg_pool2d
- Normalization Layers
  - torch.nn.functional.batch_norm
  - torch.nn.functional.instance_norm
  - torch.nn.functional.layer_norm
  - torch.nn.functional.group_norm

Submission

Project submission includes two steps.

Step 1: You need to submit your implementation on Moodle by 11/07/2024 11:59:59pm. The submission link will be closed immediately after the deadline. Your submission should be a single zip file that is named by the last names of your team members. You may use this naming structure: <groupXX_lastname1_lastname2>.zip.

The zip file should contain the following:

- A folder containing all your code.
- A simple README file that describes the structure of your code and how to run it.
- A PDF report. Use one of the provided templates to prepare your report.
  - Google Doc template: CSI 4140/5140 — Project1 Report
  - Overleaf template: https://www.overleaf.com/read/vxczqfmvpqhp#254df5

Do not include any binary file in your zip file. Only one submission is needed per group.
Failure to follow the above instructions to prepare your submission will cause a penalty to your grade.

Step 2: You need to present your project in class, 11/07/2024 1:00pm. You have 10-15 minutes for the presentation. The presentation duration may be adjusted before one week of the deadline according to the number of groups.

Policies

1) Late submissions will absolutely not be graded (unless you have verifiable proof of emergency). It is much better to submit partial work on time and get partial credit for your work than to submit late for no credit.

2) Each group needs to work independently on this exercise. We encourage high-level discussions among groups to help each other understand the concepts and principles. However, code-level discussion is prohibited and plagiarism will directly lead to failure of this course.