CSC 737: Assignment 3

Total: 100 points

Problems

1. **(85 points)** In this coding assignment you will implement and compare two CNN architectures.

   Software implementation (25 points) You need to implement the CNN models using the following steps:

   (a) Preparing the data: You will be using the CIFAR-10 dataset (https://www.cs.toronto.edu/ kriz/cifar.html). Augment the data set (only for training set) including horizontal flip, rotation (range 20), scaling (range 0.4), resizing (36) and random crop (5 pixels).

   (b) Implement plain network model: Implement a plain CNN network in the class name BaseNet. The first layer should be a Convolution layer with 16 filters of size 3×3 using same padding and stride 1. A BaseNet-20 is followed by a 3 modules where each module contains 6 Convolution layer. Each convolution layer in first module has 16 filters of size 3×3, uses same padding and a stride of 1. Each convolution layer in second module has 32 filters of size 3×3, uses same padding and a stride of 1 (except the first convolution layer which has a stride of 2). Each convolution layer in third module has 64 filters of size 3×3, uses same padding and a stride of 1 (except the first convolution layer which has a stride of 2). The final layers uses a global average pool and a dense fully connected softmax layer with 10 neurons. Each convolution layer in each module and input layer uses a Batch Normalization and ReLU activation function. A BaseNet-32 version uses 3 module in between first and last layers similar to BaseNet-20 but each module has 10 convolution layers instead of 6.

   (c) Implement Residual network model: Implement a ResNet model from scratch and name it under the class name ResNet. Similar to BaseNet it starts with a convolution layer of 16 filters of 3×3 using same padding, stride 1, Batch Normalization and ReLU activation. For ResNet-20 model, the input layer is followed by 3 residual modules. Each residual module is composed of 3 residual blocks with 16 filters in first module, 32 in second module and 64 in third module. The final layer is composed of a global average pool layer followed by a fully connected softmax layer of 10 neurons. Similar to ResNet-20, create ResNet-32 with 3 modules in between input and output layer and 5 residual blocks in each module instead of 3. You need to implement the Residual block and not call the pre-written function from the library. For reference look into the ResNet50 implementation by Francois Chollet: https://github.com/fchollet/deep-learning-models/blob/master/resnet50.py

   For each of the implementation, you are allowed to use the convolution, pooling, batch normalization, etc except for residual block, etc.

   Report (60 points) Your report should include the implementation details stating what functions you implemented in the code and the hyper parameters you are using. Use He initializer. You need to evaluate your implementation for the following experimental objectives:

   (a) **Compare the performance on layering a model** This experiment will evaluate the performance on increasing the layers in each of the architecture. Compare the training and testing accuracy over epoch for BaseNet-20 and BaseNet-32. Compare the training and testing loss over epoch for ResNet-20 and ResNet-32. Use a learning schedule where the starting learning rate is 0.1 and divide it by 10 every 200 epoch.

(b) **Compare the performance of BaseNet and ResNet** Compare the training and testing accuracy over epoch for BaseNet-20 and ResNet-20. Compare the training and testing accuracy over epoch for BaseNet-32 and ResNet-32. Compare the training and testing loss over epoch for BaseNet-20 and ResNet-20. Compare the training and testing loss over epoch for BaseNet-32 and ResNet-32. Use the learning schedule as before.

(c) **How different optimizer affect the convergence?** Compare the training accuracy and loss over epochs for BaseNet-32 and ResNet-32 model using different optimizer: learning schedule describe above, momentum and adam.

For each of the experiments state the division of testing and training data set, batch size, etc. For each experimental analysis, your report should include the objective, experimental setup including any parameter setting, plots, observation and inference statements.

2. **(15 points)** Provide a paper review for the following paper: Chen, Xinlei, and Kaiming He. "Exploring simple siamese representation learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. Link: `http://openaccess.thecvf.com/content/CVPR2021/html/Chen_Exploring_Simple_Siamese_Representation_Learning_CVPR_2021_paper.html`. Your review should include summary of the contents in the paper and critical analysis. Critical analysis should include what you like and dislike about the paper, is there any questions regarding the content. Your summary should not exceed 500 words.

General Instructions

1. Submit a single PDF or Word file for your report and paper review.
2. Submission of code need to be in code submission and report in report submission for assignment. Any cross submission will result in penalty.
3. Codes need to be implemented in Python and submitted as a single zip file in code section. Code needs to be modularized and properly documented with comments.
4. Variable naming in the code need to be descriptive or corresponding to the definition in the report.
5. Write the code and report by yourself. Do not copy code or report.
6. Do not fabricate results. If you submit results in report without software implementation to support it, your results is not corroborated and will be considered fabrications.
7. Type in your answers mostly. Do not hand-write, scan and upload.
8. All problems need to be indexed as per their indexing in the assignment.
9. Provide list of references which include books, web-sites, people and papers.
10. Do not use AI to generate your report. You can use AI as your coding guide, but need to explain your implementation details, observations and inferences.
11. No late assignment accepted.