Assignment #3

This assignment can be completed individually or by a team with up to 5 members.
Total score: 100 + 20 (bonus)
Due date: refer to the Canvas page

Objectives

To process and compute images and design multilayer backpropagation perceptron (MLP) and Convolutional Neural Network (CNN) architectures for image classification.

Only Python programs written using Python 3.0 or higher will be accepted. **NO Jupyter notebook or any other Python variants will be accepted** for efficient grading.

This assignment asks you to design and develop artificial neural network models that can classify handwritten digits from the MNIST dataset. The MNIST dataset consists of 60,000 training images and 10,000 testing images. Each image is a small 28x28 (784) pixels grayscale digit between 0 and 9. You can use a Python library to download this dataset.

Required activities

Design the **best-performing** MLP and CNN models that require **minimum computing resources** (minimum # of layers and neurons) to classify the handwritten digits and write **PyTorch** programs to implement the models.

Example programs that process images and implement MLP and CNN models using PyTorch will be posted on the course page. You can modify these programs to complete this assignment.

Warning: You are allowed to write your own PyTorch program from scratch or modify the example program but are not allowed to download and use any other programs.

Write an analysis report that describes the following elements:

(1) For your best MLP model, (a) # the of layers, (b) the total number of neurons, (c) activation function(s), (d) loss function(s), (e) gradient methods (Adagrad, Adadelta, RMSprop, Adam), (f) hyperparameters (learning rate, momentum), (g) training methods (pretraining, sequential training, batch training, SGD, minibatch), (h) training time, (i) the percentage accuracy, and (j) brief justification of why your choice of design and parameters are the best for your model.

(2) For your best CNN model, (a) # of convolutional layers, (b) the kernel (filter) size, (c) the stride, (d) # of filters, (e) # of pooling layers, (f) the pooling size, (g) pooling method (min, max, avg), (h) the number of layers and neurons for the fully connected layer, (i) activation function(s), (j) loss function(s), (k) gradient methods (Adagrad, Adadelta, RMSprop, Adam), (l) hyperparameters (learning rate, momentum), (m) training methods (pretraining, sequential training, batch training, SGD, minibatch), (n) training time, (o) the percentage accuracy, and

(p) brief justification of why your choice of design and parameters are the best for your model.

(3) A brief description of your own conclusion from the study results and recommendation of the design and parameters, if any.

Additional work for 20% bonus points

(4) Collect some images of the trees or plants from CSUF Arboretum, create a training data set, use a pretrained CNN model such as ResNet to classify some testing images of trees or plants, and write a brief report including all the elements required for the task (2).

Warning: Although you can reuse any source codes available on the Internet, you cannot share your codes with any other team or students in class. Any student or team violating this policy will receive a **ZERO** score for this assignment, potentially for all the remaining assignments.

What to submit

- An analysis report with your member names and % contribution made by each member in **PDF** or **Word format**. If every member contributed equally, state "equal contribution." If your team does not agree on individual contributions, briefly write a task description for each member. Different grades may be assigned based on individual contributions.
- Python program files
- If the data is too big, select only a few examples and include them in the report.
- DO NOT submit any zip file. Instead, upload individual files.
- Submit only one report and the program files for each team.

Grading criteria

- The overall quality of work based on the modeling results, analysis process, methods used, and source codes
- The level of understanding shown in the report
- Effort