CSE485 Deep Learning: Major Task

Leaf Classification

Part I: Data Preparation

In this project, you will use the Leaf Classification dataset using a neural network architecture.

You can use keras/tensorflow/pyTorch or write your own routines.

 1) First, download the *data* file, load it, and

   1. Describe the data
   2. Clean the data
   3. Check the data for missing values or duplicates and carry out proper correction methods
   4. Visualize the data using proper visualization methods.
   5. Draw some of the images
   6. Carry out required correlation analysis

 2) Divide the data into a training and test set using approximately 80% for training.
 3) Decide if you need to standardize the data, by computing the mean and standard deviation for each feature dimension using the training set only, then subtracting the mean and dividing by the stdev for each feature and each sample.
 4) Encode the labels

Part II: Training a neural network

In this project, you need to implement a CNN model which will be used to classify the data in Part I.

You also need to write the training function (training), and should explore the following hyperparameter settings:

- Batch size: Number of examples per training iteration.

- Number of layers: Try using different number of layers in your model and compare the performances.

- Dropout is an effective strategy to defend against overfitting. Try using different dropout rate to compare the performances.

- Optimizer: Try using different optimizers such as SGD, Adam, RMSProp.

- Regularization (weight decay): L2 regularization can be specified by setting the weight_decay parameter in optimizer. Try using different regularization factor and check the performance.

- Learning rate, Learning rate scheduler: Learning rate is key hyperparameter in model training, and you can gradually decreasing the learning rate to further improve your model. Try using different learning rate and different learning rate scheduler to compare the performance.

You could analyze one hyperparameter at a time (i.e. fixing all others to some reasonable value), rather than performing grid search. If you use TensorBoard to monitor your training, you can directly attach the screenshots of the training curves (accuracy) in your report.

To evaluate the performance of trained model, you also need to write a function (evaluation) which loads the trained model and evaluate its performance on train/test set. In your report, please clearly state what hyperparameters you explored, and what accuracy the model achieved on train/test set.

Deliverables:

- Documentation (PDF file) which includes the problem definition, screenshots of the code, screenshots of the output, visualization of the accuracy change, the description of the used model, full analysis of your results.

- A notebook file (ipynb)

- You can make a group up to 3 students.

- Submission should be in LMS

- Deadline: 24 December 2023.