ES 413: Deep Learning

Assignment 0: Feed Forward Networks

Total Marks: 10

Deadline: 2nd Feb 2023, 23:59

Implement a Multilayer Perceptron from the template provided for the 102 Flowers dataset. Plot the learning rate, training loss, and validation loss at each step and epoch (Tensorboard and Wandb are two possible options).
Save the model every 1000 steps. The code should be able to run on CPU and (single) GPU by specifying the dev_id parameter.
[5M for plots + 4M for correct implementation + 1M for hyperparameter tuning]

```python
import argparse
import torch.nn as nn

from tqdm import tqdm as tqdm

class FeedForwardNetwork(nn.Module):
   def __init__(self, input_size, output_size, layerwise_hidden_dims, layerwise_activations):
       """layerwise_hidden_dims is a list of hidden dimension size in consecutive layers.
layerwise_activations is a list that stores the activation function to be used after each layer."""
       pass

   def forward(self, x):
       pass


def model_trainer(training_args):
   """Train the model using the training arguments provided. """

   # Load the 102 flowers dataset and flatten the 3d array to 1d (average across color channels and
flatten the HxW matrix)
   train_loader = None
   val_loader   = None

   loss_criterion = None
   optimizer      = None
   ff_model       = None

   for epoch in tqdm(training_args['num_epochs'], desc="Epochs"):
       pass

   return
```

```
def parse_args():
   # Some training arguments are defined below. Feel free to define others as required.
   parser = argparse.ArgumentParser()
   parser.add_argument('--model_dir', help='name of the directory to save the model')
   parser.add_argument('--dev_id', help='cuda device id, -1 for CPU')
   parser.add_argument('--num_epochs', help='number of epochs')
   parser.add_argument('--batch_size', help='batch size')
   parser.add_argument('--lr', help='learning rate')

   args = parser.parse_args()
   return args


if __name__ == "__main__":
   training_args = parse_args()
   model_trainer(training_args=training_args)
```