Assignment (2)

24-788 Deep Learning for engineers

Out Date: 2024/1/23 (Tue)
Due Date: 2024/1/31 (Wed) @ 11:59 pm EST

All exercises should be submitted to Gradescope under **Homework 2**. The jupyter notebook should include the structure of your model, the hyper-parameters, visualization of your training process and the final accuracy/performance you achieve. Please restart and run all cells before submission. Notebooks that have not done this prior to submission will lose points. You need to save your completed notebook as a PDF for submission. This can be done with ctrl + p or command + p when you have finished running every cell. Additionally for this assignment, you should submit a zip file of the code and model weights separately to gradescope as the following file structure:

andrewID-HW2

- **p1.ipynb**
- **p1.pdf**
- **p1_model.pt**

Please use Piazza for any questions about the assignment. You can refer to Python3 tutorial, Numpy documentation and PyTorch documentation while working on this assignment. Any deviations from the submission structure shown below would attract penalty to the assignment score.

Programming Exercises (50 points)

PROBLEM 1

MLP and Back Propagation (25 points)

In this section, you are asked to implement a Multi-Layer Perceptron using PyTorch. Please follow the given template `p1.ipynb` to build your own code.

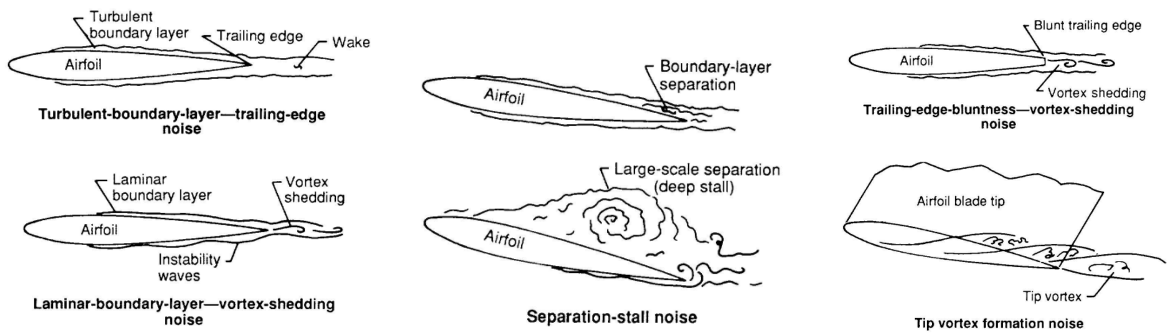

Fig. 1: Airfoil Self-Noise

The problem you are asked to solve is Airfoil Self-Noise prediction[1, 2]. Namely, given 5 features (Frequency in Hertz, Angle of attack in degrees, Chord length in meters, Free-stream velocity in meters per second, and Suction side displacement thickness in meters), your model is supposed to accurately predict the Scaled sound pressure level, in decibels. The data has been preprocessed for you and is provided in the directory as ‘train_data.npy‘, ‘validation_data.npy‘, and ‘test_data.npy‘. The classes/functions your are supposed to implement are as follows:

a) Data handling. You will implement the `__init__`, `__len__`, and `__getitem__` functions of the `Dataset` class. Your `Dataset` object will be passed into the `DataLoader` object that handles batching and shuffling between training epochs. Note: your `Dataset` object should have separate attributes for the features and targets and should return the specified feature and target as separate values in `__getitem__`. The first 5 columns of the data matrix are the features (frequency, angle of attack, chord length, free-stream velocity, suction side displacement thickness, and scaled sound pressure), while the final column is the target. (5 points).

b) Neural Network model. You will implement `NeuralNetwork` class's `__init__()` and `forward` functions. This is where you will define the model's input, hidden, and output layers, the activation function, and how the data passes through them. (5 points).

c) The network's hyperparameters: epochs, learning rate, weight decay, batch size, and loss function. (5 points).

d) Train your MLP (5 points): Here's the most exiciting section! After you finish the codes concerning building MLP model and defining loss criterion, you are asked to actually train and test your MLP model using the `train` function.

e) After training, you should plot your training and validation curves, and report your final test error in your submission and credits for this section will be graded on this. Please print the final test error towards the bottom of your notebook after model training. Note: only the best results you obtain will count towards this. (5 points).

- Error < 10: 10 points (including 5 extra points)
- $10 \leq$ Error < 15: 5 points
- $15 \leq$ Error < 20: 3 points
- Error ≥ 20: 0 points

Fig. 2 shows an example of what you should get after your code successfully trains on the Airfoil Self-Noise dataset. Note that you are not required to get the same number for loss/error during your training. The figure is only to illustrate what the plot should look like.

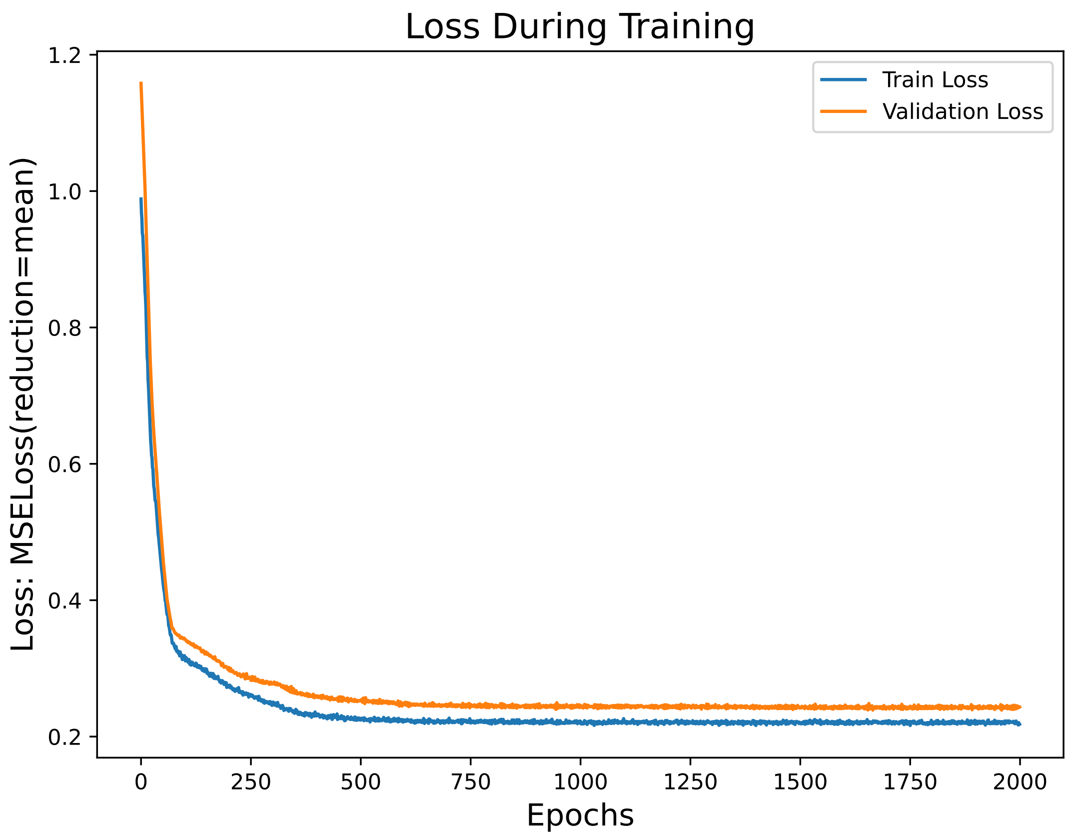

Fig. 2: Example of training plot

f) Please provide plots for 4 different combinations of hyperparameters (including varying network depths) and explain why the results differ for different hyperparameters. Explanations should be written in a markdown cell. (5 points)

You should submit your code as `hw2.ipynb`, and your trained model saved as `hw1_model.pt`. Please restart and run all cells before submission. Points will be deducted if you do not do this.

References

[1] Thomas F. Brooks, D. Stuart Pope, and Michael A. Marcolini. Airfoil self-noise and prediction. NTRS Author Affiliations: PRC Kentron, Inc., Hampton, NASA Langley Research Center NTRS Report/Patent Number: L-16528 NTRS Document ID: 19890016302 NTRS Research Center: Legacy CDMS (CDMS). July 1989. url: https://ntrs.nasa.gov/citations/19890016302 (visited on 01/24/2023).

[2] Roberto Lopez. Airfoil Self-Noise Data Set. Mar. 2014. url: https://archive.ics.uci.edu/ml/datasets/airfoil+self-noise.