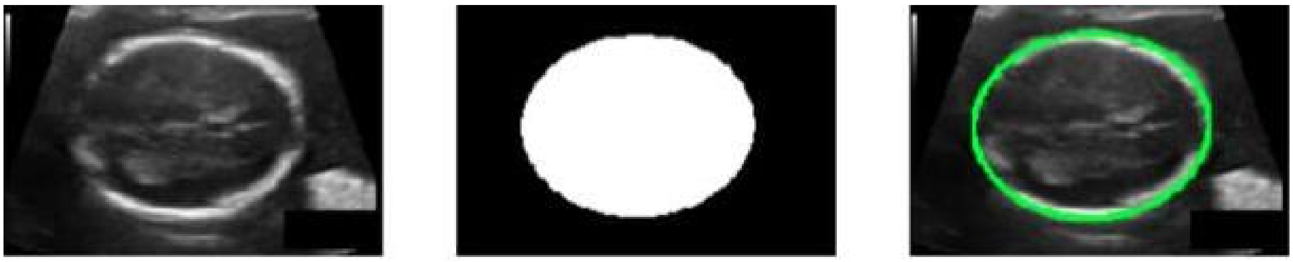
ID6001W: Applied Deep Learning
Programming Homework 4 & 5

To be submitted by April 4th, 2023 11:59 pm

1 CNN in PyTorch

1. This entire assignment must be completed in a jupyter notebook. Please submit the ipynb notebook for evaluation.
2. In this assignment, we are going to implement a CNN in Pytorch and use the same to segment out the fetal head.
3. This assignment is on developing a deep-learning model to perform single-object segmentation.In single-object segmentation, we are interested in automatically outlining the boundary of one target object in an image. The object boundary is usually defined by a binary mask. From the binary mask, we can overlay a contour on the image to outline the object boundary. For example, the following screenshot depicts an ultrasound image of a fetus, a binary mask corresponding to the fetal head, and the segmentation of the fetal head overlaid on the ultrasound image:



4. You have to use some encoder-decoder algorithm to automatically segment a fetal head in ultrasound images.

   Please download the dataset from the following link:
5. To download the dataset, visit website and and go through the following steps:

   website- https://zenodo.org/record/1322001#.ZBgHttJBy0l

   (a) Download the training_set.zip and test_test.zip files.
   (b) Extract them as training_set and test_set, respectively.

The training_set folder should contain 1,998 ṗng files, including 999 images and 999 annotations. In addition, the test_set folder should contain 335 ṗng images. There are no annotation files in the test_set folder.

1. Create the custom dataset(Split the training_set into 799 images for training and validate with 200 images).
2. Use a custom encoder-decoder model and experiment(you are allowed any set of combinations like any number of layers in the encoder-decoder, any activation function,any optimizer,any lr, etc.) to segment out the fetal head. Dice score is the metrics to be used.
3. Now use a plain UNet model and calculate the Dice score. Compare the Dice score of the models and comment which one performs better.
4. Plot the training and validation losses with the number of epochs and also plot the train-val metrics plot.
5. Deploy the model on the test_set. Also display the images and the predicted output of the model on test_set.

2 RNN in PyTorch

1. Load the required libraries such as Pandas, NumPy, Matplotlib, Pytorch and Sklearn.
2. Download the dataset from the given link and read it into a Pandas dataframe.
3. Do a train-validation-test split of the data. The recommended split is 70-15-15, i.e., 70% of the data for training, 15% for validation, and 15% for testing.
4. Create a new data frame where the features are the previous 100 values, and the label is the current value. This can be done using the Pandas shift() function
5. Train a recurrent neural network (RNN) on the created data frame. We can use Pytorch, a high-level neural network library for Python, to create and train the RNN.
6. Evaluate the performance of the trained RNN by calculating the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) values for the predictions on the test set. These metrics can be calculated using Sklearn's mean_squared_error(), mean_absolute_error(), and r2_score() functions, respectively.
7. Plot the learning curves for the RNN. This can be done using Matplotlib to plot the training and validation loss against the number of epochs.
8. Instead of using the previous values as features, use the hour, day, day of the week, week of the year information, as well as the periodicity information, along with checking whether the day is a holiday using the holiday module as features to predict the current value using RNN. This can be done by creating a new dataframe with these features and labels.
9. Train an RNN on the new dataframe using the same architecture as before
10. Evaluate the performance of the trained RNN by calculating the MSE, RMSE, and R^2 values for the predictions on the test set.
11. Plot the learning curves for the new RNN.
12. Compare the performance of the two RNNs and determine which feature engineering method is better for the given dataset.
13. Submit the Jupyter notebook for evaluation.