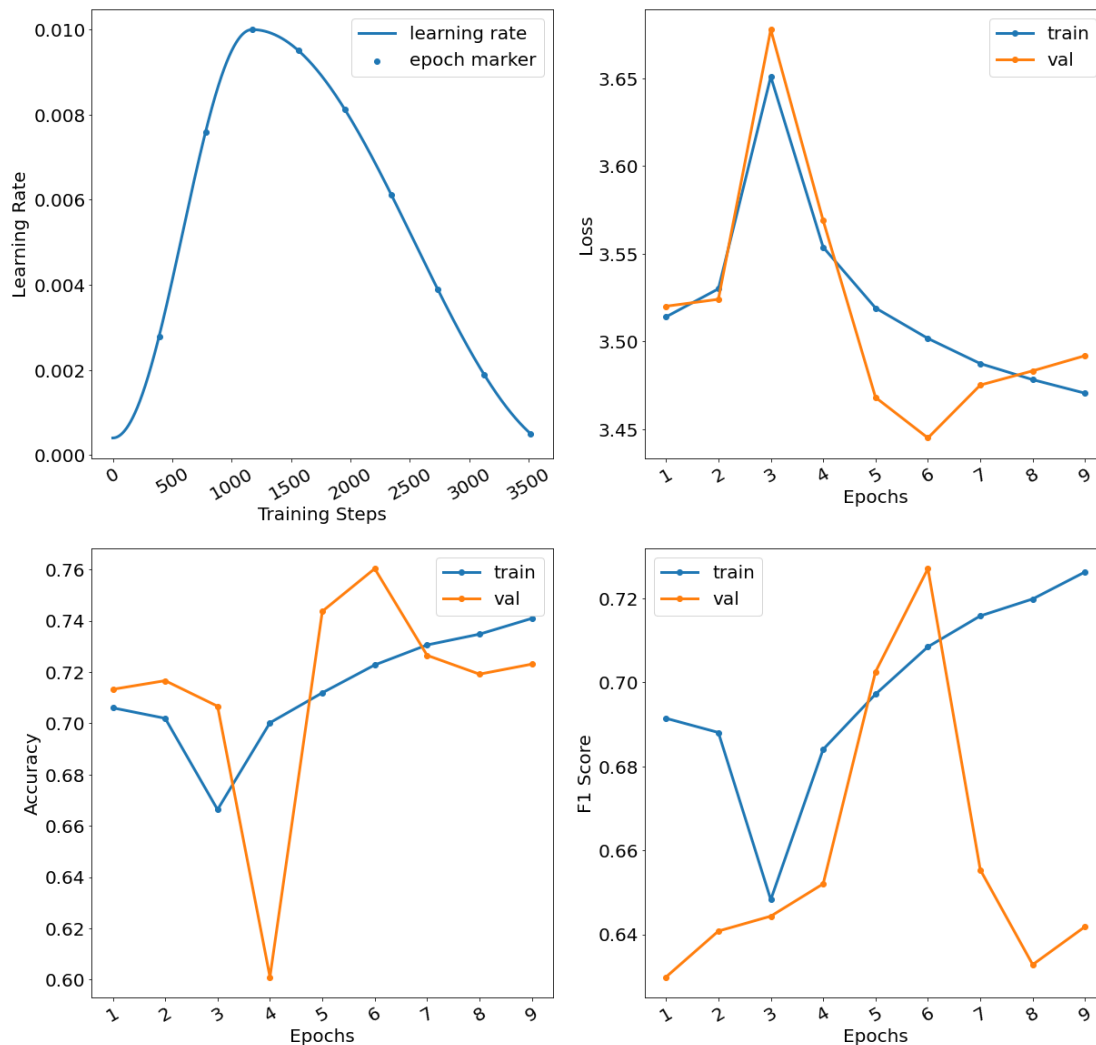


Assignment 1 – “Hello World”

Description:

For this course, you will have many hands-on experiences in developing machine learning algorithms and writing code. The programming language we are going to use is Python. For this assignment, you need to use Python to draw a plot to show the training and validation performance of a neural network model, as shown in the figure below. The training and validation results are given to you as *.npy* files.



To complete this assignment, you need to set up the programming environment first, which is the main purpose of this assignment. I hope this assignment will help you set up your own development environment and get ready for this course.

Directions:

There are many Python development environments available. You may choose to use any of those. But I would suggest [Google Colab](#), which is provided by Google. It could be considered the Google version of the Jupyter Notebook. In addition to the regular Jupyter Notebook functions, Google Colab also provides free GPU hours to users, which could be extremely helpful for some of our later assignments.

If you choose to use Google Colab, you need to follow the following instructions:

- Create a Google account if you don't have one
- Go to your Google Drive
- Create a new Google Colab file
 - Click "New" in the top-left corner
 - If you cannot see Google Colaboratory in the drop-down menu, click "More"
 - If you still cannot see Google Colab, click "Connect more apps."
 - Search for "Colaboratory"
 - Click "Colaboratory"
 - Install "Colaboratory"
- In the Colab file:
 - Use *numpy* to load the two given .npz files.
 - Use Matplotlib to draw the plots

If you choose to use other development environments, you may:

- Open the environment at your choice
- Use *numpy* to load the two given .npz files.
- Use Matplotlib to draw the plots

Data File and Plotted Result:

Three .npz files are given: 1) *learning_rate.npz*, 2) *train_hist.npz*, and 3) *val_hist.npz*.

- *learning_rate.npz*
 - The shape of this file is (3528, 1).
 - Each row shows the learning rate at a specific training step.
 - The model is trained for 3528 steps.
- *train_hist.npz* and *val_hist.npz*
 - These two files are for the training result and validation result, respectively.
 - The training and validation results are done using two different datasets. You don't need to worry much about the real meanings of this assignment.
 - The model was trained for 9 epochs. After each epoch, we recorded the current performance using three metrics: 1) loss, 2) accuracy, and 3) F1 score.
 - "Epoch" is a term used to describe neural network training length. You don't need to worry about the meaning of "epoch" for this assignment.
 - For this specific model, one epoch equals 391 training steps.
 - Note the *learning_rate.npz* contains data for 5329 training steps, which is 9 epochs + 3 more steps.
 - Each of the *hist* files contains a 9-by-3 matrix.
 - Each row is for the result after an epoch.
 - The three columns are for 1) loss, 2) accuracy, and 3) F1 score, respectively.

- E.g., 2nd row for the *train_hist.npy* file shows the loss, accuracy, and F1 score of the training performance after the 2nd epoch.

For this assignment, you need to load the three *.npy* files and plot the result as shown at the beginning of this file. Below are the specific requirements for the plots:

- The figure contains 4 subplots
- You want to plot the learning rate at Row1 Column 1
 - The x-axis shows the training steps
 - The y-axis shows the learning rate value at a given step
 - Rotate the ticks of the x-axis by 30 degree
 - Set the x-axis label as “Training Step” and the y-axis label as “Learning Rate”
 - Use “•” to mark the last training step of each epoch (i.e., 391, 782, 1183, etc.).
 - Add legends to show that the line means “learning rate” and dots are the “epoch makers.”
 - You may use *Matplotlib.plot* to draw the line and *Matplotlib.scatter* to draw the dots.
- Plot the Loss, Accuracy, and F1 Score at Row1Column2, Row2Column1, and Row2Column2, respectively.
 - For each plot, the x-axis shows the epoch, and the y-axis shows the corresponding values.
 - Rotate the ticks of the x-axis by 30 degree
 - Set the x-axis label as “Epochs” and the y-axis label as “Loss,” “Accuracy,” and “F1 Score,” respectively.
 - Plot the training and validation results of the same metric in the same plot. Use legends to indicate which line is for training and which line is for validation.
 - If using *Matplotlib.plot* to draw the lines, set line style as “-o” to draw the solid line with dots.

Submission:

Make a screenshot of your code and your plots. Submit the screenshot in Blackboard. Make sure the screenshot is readable. If necessary, feel free to upload multiple screenshots.

Evaluation Criteria:

This assignment is worth 100 points.

- 30 points for participation
- 70 points for the correctness of the code and drawing
 - 4 subplots 20 points
 - Correct plots 20 points
 - Legends 10 points
 - Correct axes and labels 10 points
 - Rotate x-axis ticks 10 points