Tracking a Model building experiment in MLFlow

We will build several neural network models for MNIST digit classification with different model configuration settings. Each configuration setting will show different pattern in model building performance measurements over every learning epoch. We want to compare the model performance trends and understand the impact of model configuraion settings.

Manually coding for data collection and visualization is a pain. MLFlow comes to our help here where tracking the "metrics" and "parameters" is automatically done by MLFlow through the `mlflow.autolog()` function call in your model building script.

Start with the jupyter notebook (MNIST.ipynb) provided to you which has all the necessary code for building the classification model with different parameter settings. Your objective is setup MLFlow in your machine and add MLFlow logging to the MNIST model building notebook followed by ensuring that the experiments are tracked on the MLflow console.

There are at least 10 variations of the network in the notebook provided. This should translated to 10 different experiments in MLFlow console.

Tasks

1. Modify the code to log metrics (loss, accuracy) etc. You may use mlflow.autolog() and compare it with mlflow.log_metrics(). [10 pts]
2. Modify the code to log paramters (network configuration, learning rate, optimizer, regularization, etc). You may use mlflow.autolog() and compare it with mlflow.log_param(). [10 pts]
3. Use the "with mlflow.start_run()" construct to run the model build for each configuration. This will create an entry in MLflow. Each time your rerun the code block, a new entry shall be created in MLflow. [15 pts]
4. Let the entire MNIST experiment (superset of 10 variants) be configured as an experiment. So, each neural configuration will become a sub-experiment under.
    a) Refer to the "hyperparameter search" notebook that was used in the class to learn about sub-experiments being grouped under the full-experiment.
5. When you run the notebook in full, you should see one entry in MLflow with [+] sign and upon clicking on [+] should expand to 10 sub-experiments. [5 pts]
6. Compare the performance numbers using the "Compare" button and take snapshots of the comparison plots. Note that you are comparing within one group of sub-models. [5 pts]
7. Also compare models across experiments and take snapshots of the plots. [5 pts]

Important Pointers

1. You should strictly follow the function prototypes when you build the functions.
2. You are expected to submit notebook files and the snapshots of your mlflow console.
3. Here is the allocation of points per task
   10% for a clean coding style
   40% for the correctness of the implementation
   20% for readable comments
   10% for input arguments' validation/boundary check
   20% for unit test modules

Best wishes.