twinlab.Emulator.learn

Emulator.learn(*dataset, inputs, outputs, num_loops, num_points_per_loop, acq_func, simulation, train_params=<twinlab.params.TrainParams object>, recommend_params=<twinlab.params.RecommendParams object>, verbose=True*)

Perform active learning to improve an emulator on the twinLab cloud.

Active learning is a method that can identify and utilise the most informative data points to add to an emulator in order to reduce the number of measurements to be taken or simulations that are required. Using active learning can result in a more accurate model, trained with less data. The primary difference between this method and `Emulator.recommend` is that in this method, the emulator is trained, new data points are suggested, and then training occurs continuously in an active loop. This way, new data can be used to train and update an emulator until the desired level of accuracy is achieved. This can be done using either the `"optmise"` or `"explore"` acquisition functions. The emulator is therefore updated on the twinLab cloud with the objective of either finding the point of maximum output or reducing the overall uncertainty in the emulator. This method does not return anything to the user directly, but instead updates the `Dataset` and `Emulator` in the cloud.

Parameters:

- **dataset** (*Dataset*) – twinLab dataset object which contains the initial training data for the emulator.
- **inputs** (*list[str]*) – List of input column names in the training dataset.
- **outputs** (*list[str]*) – List of output column names in the training dataset.
- **num_loops** (*int*) – Number of loops to run of the learning process. This must be a positive integer. Note that in this method, the emulator is trained and then re-trained on new suggested data points, so setting `num_loops=1` here will mean that `Emulator.train` is run _twice_, and `Emulator.recommend` is run once.
- **num_points_per_loop** (*int*) – Number of points to sample in each loop.
- **acq_func** (*str*) – Specifies the acquisition function to be used when recommending new points: either `"explore"` or `"optimise"`.
- **simulation** (*Callable*) – A function that takes in a set of inputs and generates the outputs (for example, a simulator for the data generating process).
- **train_params** (*TrainParams, optional*) – A parameter configuration that contains optional training

running a learning loop.

- **recommend_params** (*RecommendParams, optional*) – A parameter configuration that contains optional recommendation parameters.
- **verbose** (*bool, optional*) – Display detailed information about the operation while running. If `True`, the requested candidate points will be printed to the screen while running. If `False` the emulator will be updated on the cloud while the method runs silently.

Return type:

`None`

Examples

```
emulator = tl.Emulator("quickstart")
dataset = tl.Dataset("quickstart")
emulator.learn(
    dataset=dataset,
    inputs=["x"],
    outputs=["y"],
    num_loops=3,
    num_points_per_loop=5,
    acq_func="explore",
    simulation=my_simulator,
)
```

Previous
twinlab.Emulator.recommend

Next
twinlab.Emulator.calibrate