

# twinlab.Emulator.train

**Emulator.train**(*dataset*, *inputs*, *outputs*, *params*=<*twinlab.params.TrainParams* object>, *wait*=True, *verbose*=False)

Train an emulator on the twinLab cloud.

This is the primary functionality of twinLab, where an emulator is trained on a dataset. The emulator learns trends in the dataset and then is able to make predictions on new data. These new data can be far away from the training data, and the emulator will interpolate between the training data points. The emulator can also be used to extrapolate beyond the training data, but this is less reliable. The emulator can be trained on a dataset with multiple inputs and outputs, and can be used to make predictions on new data with multiple inputs and outputs. The powerful algorithms in twinLab allow for the emulator to not only make predictions, but to also quantify the uncertainty in these predictions. This is extremely advantageous, because it allows for the reliability of the predictions to be quantified.

## Parameters:

- **dataset** ([Dataset](#)) – The training and test data for the emulator. The ratio of train to test data can be set in `TrainParams`.
- **inputs** ([list\[str\]](#)) – A list of the input column names in the training dataset. These correspond to the independent variables in the dataset, which are often the parameters of a model. These are usually known as `X` (note that capital) values.
- **outputs** ([list\[str\]](#)) – A list of the output column names in the training dataset. These correspond to the dependent variables in the dataset, which are often the results of a model. These are usually known as `y` values.
- **params** ([TrainParams](#), *optional*) – A training parameter configuration that contains all optional training parameters.
- **wait** ([bool](#), *optional*) – If `True` wait for the job to complete, otherwise return the process ID and exit. Setting `wait=False` is useful for longer training jobs.
- **verbose** ([bool](#), *optional*) – Display information about the operation while running.

## Return type:

`Optional[str]`

## Returns:

If `wait=True` the function will run until the emulator is trained on the cloud. If `wait=False` the function will return the process ID and exit. This is useful for longer training jobs. The training status can then be checked later using `Emulator.status()`.

## Example

---

```
df = pd.DataFrame({"X": [1, 2, 3, 4], "y": [1, 4, 9, 16]})
dataset = tl.Dataset("my_dataset")
dataset.upload(df)
emulator = tl.Emulator("my_emulator")
emulator.train(dataset, ["X"], ["y"])
```

< Previous  
[twinlab.Emulator.design](#)

Next >  
[twinlab.Emulator.status](#)