Pyinfer

Release 0.0.1

Charles Pierse

Feb 19, 2021

CONTENTS:

Pyinfer is a model agnostic tool for ML developers and researchers to benchmark the inference statistics for machine learning models or functions.

Find Pyinfer on Github

CHAPTER

ONE

INSTALLATION

```
pip install pyinfer
```

CHAPTER

TWO

OVERVIEW

Inference Report

`InferenceReport` is for reporting inference statistics on a single model artifact. To create a valid report simply pass it a callable model function or method, valid input(s), and either **n_iterations** or **n_seconds** to determine what interval the report uses for its run duration. Check out the docs for more information on the optional parameters that can be passed.

```
from pyinfer import InferenceReport

model = MyAmazingMLModel()

# test model over 1000 iterations
report = InferenceReport(model=my_model.predict,inputs=6,n_iterations=1000)

report.run()

# test model over 10 seconds
report = InferenceReport(model=my_model.predict,inputs=6,n_seconds=10)

report.run()

# plot the model runs against run time
report.plot()
```

Multi Inference Report

`MultiInferenceReport` is for reporting inference statistics on a list of model artifacts. To create a valid multi report pass it a list of callable model functions or methods, a list of valid input(s), and either **n_iterations** or **n_seconds** to determine what interval the report uses for its run duration. Check out the docs for more information on the optional parameters that can be passed.

```python
from pyinfer import MultiInferenceReport

model1 = MyAmazingMLModel()
model2 = MyOtherMLModel()
model3 = MyOtherAmazingMLModel()

# test all models over 1000 iterations each (3000 iterations overall)
multi_report = MultiInferenceReport(
    models=[model1.predict, model2.predict, model3.predict],
    inputs=[1,2,3],
    n_iterations=1000,
    model_names=["amazingMlModel","otherMlModel","otherAmazingMlModel"])

multi_report.run()

# test all model over 10 secondseach (30 seconds overall)
multi_report = MultiInferenceReport(
    models=[model1.predict, model2.predict, model3.predict],
    inputs=[1,2,3],
    n_seconds=10,
    model_names=["amazingMlModel","otherMlModel","otherAmazingMlModel"])

multi_report.run()

# plot the model runs against run time
multi_report.plot()
```

2.1 InfererenceReport

class pyinfer.**InferenceReport**(*model: Callable*, *inputs: Any*, *n_seconds: Optional[Union[int, float]] = None*, *n_iterations: Optional[int] = None*, *exit_on_inputs_exhausted: bool = False*, *infer_failure_point: Optional[Union[int, float]] = None*, *model_name: Optional[str] = None*, *drop_stats: Optional[List[str]] = None*)

A model agnostic report of inference related stats for any callable model

__init__(*model: Callable*, *inputs: Any*, *n_seconds: Optional[Union[int, float]] = None*, *n_iterations: Optional[int] = None*, *exit_on_inputs_exhausted: bool = False*, *infer_failure_point: Optional[Union[int, float]] = None*, *model_name: Optional[str] = None*, *drop_stats: Optional[List[str]] = None*)

Parameters

- **model** (*Callable*) – The callable method or function for the model.
- **inputs** (*Any*) – The input(s) parameters the model receives.
- **n_seconds** (*Union[int, float, None], optional*) – Number of seconds to run model inferences. If this is *None* it is expected that *n_iterations* will be set. Defaults to None.
- **n_iterations** (*int, optional*) – Number of iterations to run model inferences for. If this is *None* it is expected that *n_seconds* will be set. Defaults to None.
- **exit_on_inputs_exhausted** (*bool, optional*) – If inputs are a iterable of inputs exit on completion. This feature is not yet implemented. Defaults to False.
- **infer_failure_point** (*Union[int, float, None], optional*) – Time in seconds (int or float) at which an inference is to be considered a failure in the reporting stats. Defaults to None.
- **model_name** (*str, optional*) – The name to give to the model for the report. Defaults to None.
- **drop_stats** (*List[str], optional*) – List of keys to drop from the report. Defaults to None.

Raises

- **ModelIsNotCallableError** – Will raise if the model provided is not callable.
- **MeasurementIntervalNotSetError** – Will raise if neither *n_seconds* or *n_iterations* are set.

run (*print_report: bool = True*) → dict

Runs the inference report for *self.model* with input(s) *self.inputs*

Parameters

- **print_report** (*bool, optional*) – If true a table representation of the report will be
- **to console. Defaults to True.** (*printed*) –

Returns A dictionary containing all the report stats created during the run.

Return type dict

report (*results_dict: dict*)

Prints a report to console based on the values found in *results_dict*

Parameters **results_dict** (*dict*) – Dictionary containing compiled stats from a run.

plot (*show: bool = True*, *save_location: Optional[str] = None*)

Creates a simple plot of *self.runs*. Plots run number on the x-axis and run time in milliseconds on the y-axis.

Parameters

- **show** (*bool, optional*) – Whether to show the plot after calling method. Defaults to True.
- **save_location** (*str, optional*) – Location to save plot at. If None the plot will not be saved. Defaults to None.

Raises

- **MatplotlibNotInstalledError** – Raise if matplotlib is not installed in python environment.

- **ValueError** – Raise if the runs have not yet been calculated but *plot* is called.

2.2 MultiInferenceReport

class pyinfer.**MultiInferenceReport**(*models: List[Callable]*, *inputs: List[Any]*, *n_seconds: Optional[Union[int, float]] = None*, *n_iterations: Optional[int] = None*, *exit_on_inputs_exhausted: bool = False*, *infer_failure_point: Optional[Union[int, float]] = None*, *model_names: Optional[List[str]] = None*, *drop_stats: Optional[List[str]] = None*)

A model agnostic report of inference related stats for any list of callable models

__init__(*models: List[Callable]*, *inputs: List[Any]*, *n_seconds: Optional[Union[int, float]] = None*, *n_iterations: Optional[int] = None*, *exit_on_inputs_exhausted: bool = False*, *infer_failure_point: Optional[Union[int, float]] = None*, *model_names: Optional[List[str]] = None*, *drop_stats: Optional[List[str]] = None*)

Parameters

- **models** (*List[Callable]*) – A list of the callable methods or functions for the models.
- **inputs** (*List[Any]*) – The input(s) parameters each of the models receives. If only one input is given then it is assumed each model takes the same shape/type of input and that input will be passed to each model.
- **n_seconds** (*Union[int, float, None], optional*) – Number of seconds to run model inferences. If this is *None* it is expected that *n_iterations* will be set. Defaults to None.
- **n_iterations** (*int, optional*) – Number of iterations to run model inferences for. If this is *None* it is expected that *n_seconds* will be set. Defaults to None.
- **exit_on_inputs_exhausted** (*bool, optional*) – If inputs are a iterable of inputs exit on completion. This feature is not yet implemented. Defaults to False.
- **infer_failure_point** (*Union[int, float, None], optional*) – Time in seconds (int or float) at which an inference. is to be considered a failure in the reporting stats. Defaults to None.
- **model_names** (*List[str], optional*) – The names to give to the models for the report. Must be the same length as number of models provided. Defaults to None.
- **drop_stats** (*List[str], optional*) – List of keys to drop from the report. Defaults to None.

Raises

- **ModelIsNotCallableError** – Will raise if the model provided is not callable.
- **NamesNotEqualsModelsLengthError** – Will raise if the number of models names does not match the number of model callables provided.
- **MeasurementIntervalNotSetError** – Will raise if neither *n_seconds* or *n_iterations* are set.

run(*print_report: bool = True*) → List[dict]

Runs the multi inference report for *self.models* with input(s) *self.inputs*

Parameters **print_report** (*bool, optional*) – If true a table representation of the report will be printed to console. Defaults to True.

Returns

A list of dictionaries containing all the report stats created during the run for each model callable.

Return type List[dict]

report (*results_list: List[dict]*)
Prints a report to console based on the values found in *results_list*

Parameters **results_list** (*dict*) – A list of dictionaries containing compiled stats from the runs.

plot (*show: bool = True*, *save_location: Optional[str] = None*)
Creates a simple plot of *self.models_runs*. For each run it plots run number on the x-axis and run time in milliseconds on the y-axis.

Parameters

- **show** (*bool, optional*) – Whether to show the plot after calling method. Defaults to True.
- **save_location** (*str, optional*) – Location to save plot at. If None the plot will not be saved. Defaults to None.

Raises

- **MatplotlibNotInstalledError** – Raise if matplotlib is not installed in python environment.
- **ValueError** – Raise if the model_runs have not yet been calculated but *plot* is called.

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

INDEX

Symbols

I

M

P

R