ventu

Oct 06, 2021

Contents:

Serving the deep learning models easily.

CHAPTER 1

Install

```
pip install ventu
```

CHAPTER 2

Features

- Only need to implement Model(`preprocess`, `postprocess`, `inference` or `batch_inference`)
- request & response data validation using pydantic
- API document using SpecTree (when run with `run_http`)
- backend service using falcon supports both JSON and msgpack
- dynamic batching with batching using Unix domain socket or TCP
  - errors in one request won't affect others in the same batch
- support all the runtime
- health check
- inference warm-up

2.1 Ventu API

class ventu.ventu.**Ventu**(*req_schema*, *resp_schema*, *use_msgpack=False*, **args*, ***kwargs*)

Ventu: built for deep learning model serving

Parameters

- **req_schema** – request schema defined with `pydantic.BaseModel`
- **resp_schema** – response schema defined with `pydantic.BaseModel`
- **use_msgpack** (*bool*) – use msgpack for serialization or not (default: JSON)
- **args** –
- **kwargs** –

To create a model service, inherit this class and implement:

- `preprocess` (optional)

- `postprocess` (optional)
- `inference` (for standalone HTTP service)
- `batch_inference` (when working with batching service)

app
Falcon application with SpecTree validation

batch_inference(*batch*)
batch inference the preprocessed data

Parameters **batch** – a list of data after *preprocess*

Returns a list of inference results

health_check(*batch=False*)
health check for model inference (can also be used to warm-up)

Parameters **batch** (*bool*) – batch inference or single inference (default)

Return bool `True` if passed health check

inference(*data*)
inference the preprocessed data

Parameters **data** – data after *preprocess*

Returns inference result

postprocess(*data*)
postprocess the inference result

Parameters **data** – data after *inference* or one item of the *batch_inference*

Returns as defined in `resp_schema`

preprocess(*data*)
preprocess the data

Parameters **data** – as defined in `req_schema`

Returns this will be the input data of *inference* or one item of the input data of *batch_inference*

run_http(*host=None*, *port=None*)
run the HTTP service

Parameters

- **host** (*string*) – host address
- **port** (*int*) – service port

run_tcp(*host=None*, *port=None*)
run as an inference worker with TCP

Parameters

- **host** (*string*) – host address
- **port** (*int*) – service port

run_unix(*addr=None*)
run as an inference worker with Unix domain socket

Parameters **addr** (*string*) – socket file address

sock
socket used for communication with batching service

this is a instance of *ventu.protocol.BatchProtocol*

2.1.1 Config

Check `pydantic.BaseSettings`

class `ventu.config.`**Config**
default config, can be rewrite with environment variables begin with `ventu_`

Variables

- **name** – default service name shown in OpenAPI
- **version** – default service version shown in OpenAPI
- **host** – default host address for the HTTP service
- **port** – default port for the HTTP service
- **socket** – default socket file to communicate with batching service

2.1.2 Protocol

class `ventu.protocol.`**BatchProtocol**(*infer*, *req_schema*, *resp_schema*, *use_msgpack*)
protocol used to communicate with batching service

Parameters

- **infer** – model infer function (contains *preprocess*, *batch_inference* and *postprocess*)
- **req_schema** – request schema defined with *pydantic*
- **resp_schema** – response schema defined with *pydantic*
- **use_msgpack** (*bool*) – use msgpack for serialization or not (default: JSON)

process(*conn*)
process batch queries and return the inference results

Parameters **conn** – socket connection

run(*addr*, *protocol='unix'*)
run socket communication

this should run **after** the socket file is created by the batching service

Parameters

- **protocol** (*string*) – 'unix' or 'tcp'
- **addr** – socket file path or (host:str, port:int)

stop()
stop the socket communication

2.1.3 HTTP service

class ventu.service.**ServiceStatus**
service health status

class ventu.service.**StatusEnum**
An enumeration.

ventu.service.**create_app**(*infer*, *metric_registry*, *health_check*, *req_schema*, *resp_schema*, *use_msgpack*, *config*)
create `falcon` application

Parameters

- **infer** – model infer function (contains *preprocess*, *inference*, and *postprocess*)
- **metric_registry** – Prometheus metric registry
- **health_check** – model health check function (need examples provided in schema)
- **req_schema** – request schema defined with `pydantic.BaseModel`
- **resp_schema** – request schema defined with `pydantic.BaseModel`
- **use_msgpack** (*bool*) – use msgpack for serialization or not (default: JSON)
- **config** – configs *ventu.config.Config*

Returns a `falcon` application

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

v

Index