

Assignment - 6

Build a FastAPI for MNIST digit prediction

Name- Abhishek Raj

Roll- MM20B003

Course- Big Data Lab CS5830

Introduction :

A contemporary, quick framework for creating Python APIs is called FastAPI. Its asynchronous programming and high speed architecture enable it to manage several concurrent connections with ease. Additionally, we can run our apps on the local server via its Swagger UI. Here, we attempt to put up an API for the MNIST data and do the tests on it. Since the second model from the last assignment produced one of the greatest test accuracies, we employ it.

We apply the functions as described in the question document. On running the commands, and importing uvicorn and loading swagger UI which is a web server for running ASGI applications.: we get this interface

FastAPI 0.1.0 OAS 3.1

/openapi.json

default

POST /upload/ Create Upload File

Parameters

No parameters

Request body ^{required} multipart/form-data

file ^{required} sample 21.jpg

string(\$binary)

Servers

These operation-level options override the global server options.

/

Initially, we provide the API with a 28x28 image from MNIST, as we have not yet developed the code for using arbitrary image sizes. The output is shown in the next figure.

To use arbitrary images from MS Paint, we resize them using ``image.resize((28,28))`` to get images of normal size. We ensure that the background is black, just like the MNIST dataset, and draw accordingly. We conduct 10 experiments and obtain the following results:

- For 8, the output is correct.
- For 5, the output is correct.
- For 6, the output is correct.
- For 9, the model gets it right.
- For 3, the output is correct.
- For 0, the output is correct.
- For 5, the output is correct.
- For 1, the output is correct.
- For 2, the output is correct.
- For 7, the output is correct.

****Conclusion****

We have explored various MLflow commands, including ``start run``, ``autolog``, ``mlflow.log params``, and ``mlflow.log metrics``, as well as nested runs. We have observed the various use cases and obtained a comparison between various models used for MNIST. We find that Model 8 has slightly better overall test accuracy. We also gain insight into the variability of various metrics across various experiments, demonstrating the usefulness of MLflow in streamlining and quickening model analysis.

1.



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@Screenshot 2024-04-29 220504.jpg;type=image/jpeg'
```

Request URL

http://127.0.0.1:8000/upload/

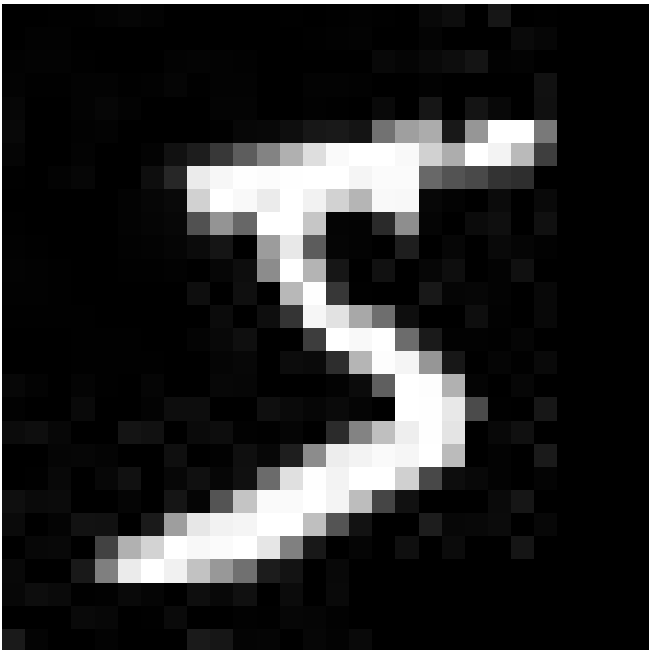
Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "digit": "8" }</pre></div><div><div>Response headers</div><pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:35:41 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	<div>Successful Response</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div>	No links

2.



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@mist_sampling.jpg;type=image/jpeg'
```

Request URL

http://127.0.0.1:8000/upload/

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "digit": "5" }</pre></div><div><div>Response headers</div><div><pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:05:58 GMT server: uvicorn</pre></div></div></div>

Responses

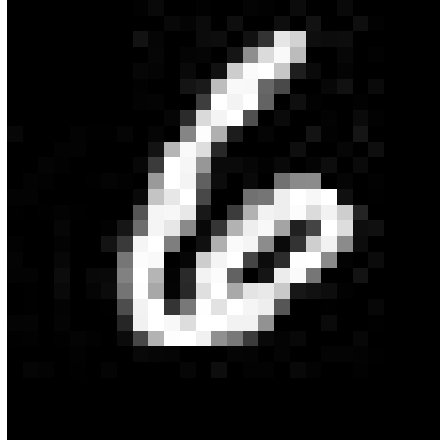
Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header

3.



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@mist_1.jpg;type=image/jpeg'
```

Request URL

```
http://127.0.0.1:8000/upload/
```

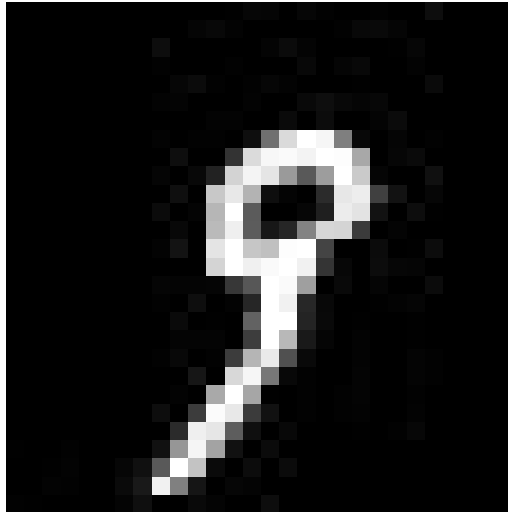
Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "digit": "6" }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:04:15 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	<div>Successful Response</div> <div><div>Media type</div><div><div>application/json</div></div><div>Controls Accept header.</div></div>	No links

4.



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@mnist_2.jpg;type=image/jpeg'
```

Request URL

http://127.0.0.1:8000/upload/

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "digit": "9" }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:14:48 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Powered by [Kong](#)

5.



/

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@mnist_3.jpg;type=image/jpeg'
```

Request URL

```
http://127.0.0.1:8000/upload/
```

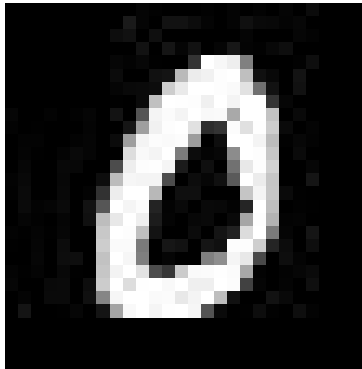
Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "digit": "3" }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:21:52 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
------	-------------	-------

6.



These operation-level options override the global server options.

/

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@sample_6.jpg;type=image/jpeg'
```

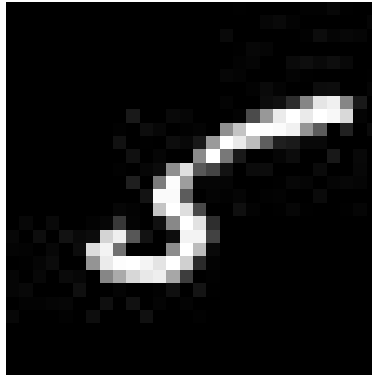
Request URL

http://127.0.0.1:8000/upload/

Server response

Code	Details
200	<p>Response body</p> <pre>{ "digit": "0" }</pre> <p>Response headers</p> <pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:32:41 GMT server: uvicorn</pre>

7.



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@sample_0.jpg;type=image/jpeg'
```

Request URL

http://127.0.0.1:8000/upload/

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "digit": "5" }</pre></div><div><div>Response headers</div><div><pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:42:50 GMT server: uvicorn</pre></div></div></div>

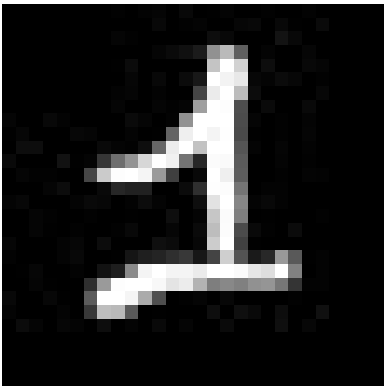
Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

8.



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@sample_9.jpg;type=image/jpeg'
```

Request URL

http://127.0.0.1:8000/upload/

Server response

Code	Details
------	---------

200

Response body

```
{
  "digit": "1"
}
```

Response headers

```
content-length: 13
content-type: application/json
date: Mon, 29 Apr 2024 16:45:39 GMT
server: uvicorn
```

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

9.



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@sample_11.jpg;type=image/jpeg'
```

Request URL

http://127.0.0.1:8000/upload/

Server response

Code	Details
200	<p>Response body</p> <pre>{ "digit": "2" }</pre> <p>Response headers</p> <pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:55:34 GMT server: uvicorn</pre>

Responses

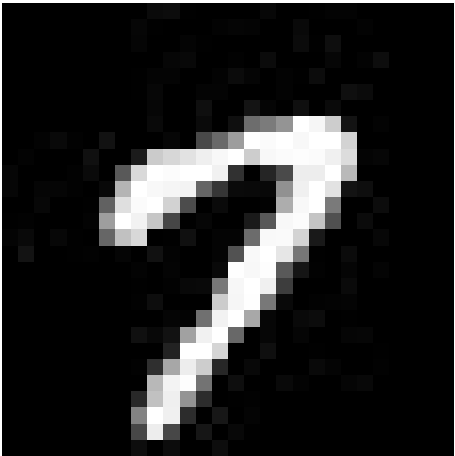
Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

10.



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@sample_12.jpg;type=image/jpeg'
```

Request URL

http://127.0.0.1:8000/upload/

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "digit": "7" }</pre></div><div><div>Response headers</div><pre>content-length: 13 content-type: application/json date: Mon, 29 Apr 2024 16:58:28 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.