Contents

1 API Documentation

For each API, we use POST to deliver form data, and use json for passing of data structures. The response is a json string.

1.1 User

1.1.1 Register a new user

POST /api/auth/register

1. Form data

Name	Format	Description
username	string	The username of the user
password	string	The password of the user (or hash of the password)

2. Response

Name	Format	Description
success	bool	Whether the register is success
uuid	string	The uuid of the user (only when success)
reason	string	Human readable reason of failure (when fail)

1.1.2 Login an existing user

POST /api/auth/login

1. Form data See POST /api/auth/register
2. Response 400 error when the username and/or password is not valid.

Name	Format	Description
uuid	string	The uuid of the user
expires	float	The timestamp when the session expires
token	string	The token of the session

1.2 Model

1.2.1 Get or modify the parameters

1. Change POST /api/model/params
   (a) Form data

Name	Format	Description
learning_rate	float	The target learning rate

   (b) Response See GET /api/model/params
2. Get GET /api/model/params
   (a) Response

Name	Format	Description
learning_rate	float	The changed (POST) or original (GET) learning rate

1.2.2 Forward the model

POST /api/model/forward

1. Form data

Name	Format	Description
x	int[]	The input data

2. Response See GET /api/model/output

1.2.3 Backward the model

GET /api/model/backward

1. Response See GET /api/model/output

1.2.4 Get the output of the model

GET /api/model/output

1. Response

Name	Format	Description
A	int	The output data

1.2.5 Optimize the model

GET /api/model/optimize

Name	Format	Description
dW	int[][]	The gradient of W
dB	int	The gradient of B

1.2.6 Get the model

GET /api/model/model

1. Response

Name	Format	Description
W	int[][]	The weights matrix
B	int	The bias