

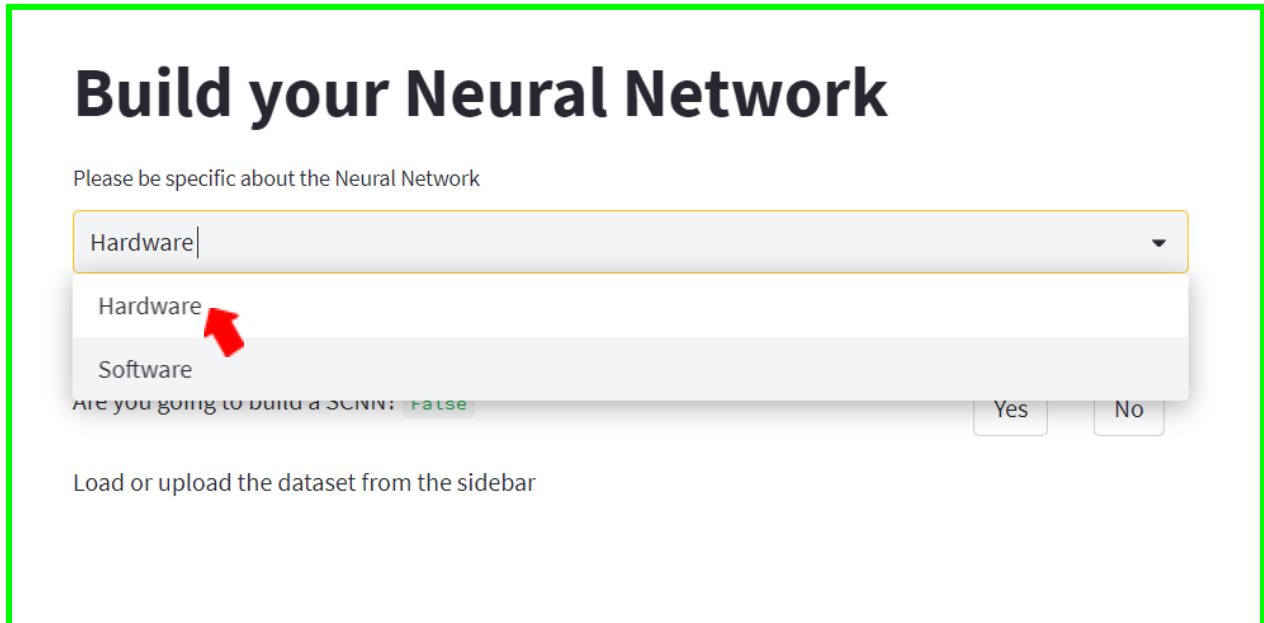
# **Website User Manual**

**Memristor-Based Neural Search**

**Optimization GUI**

## Step 1

First, the user can make the decision regarding the desired subject on which the neural network will be constructed.



**Build your Neural Network**

Please be specific about the Neural Network

Hardware

Hardware  
Software

Are you going to build a SCNN? False

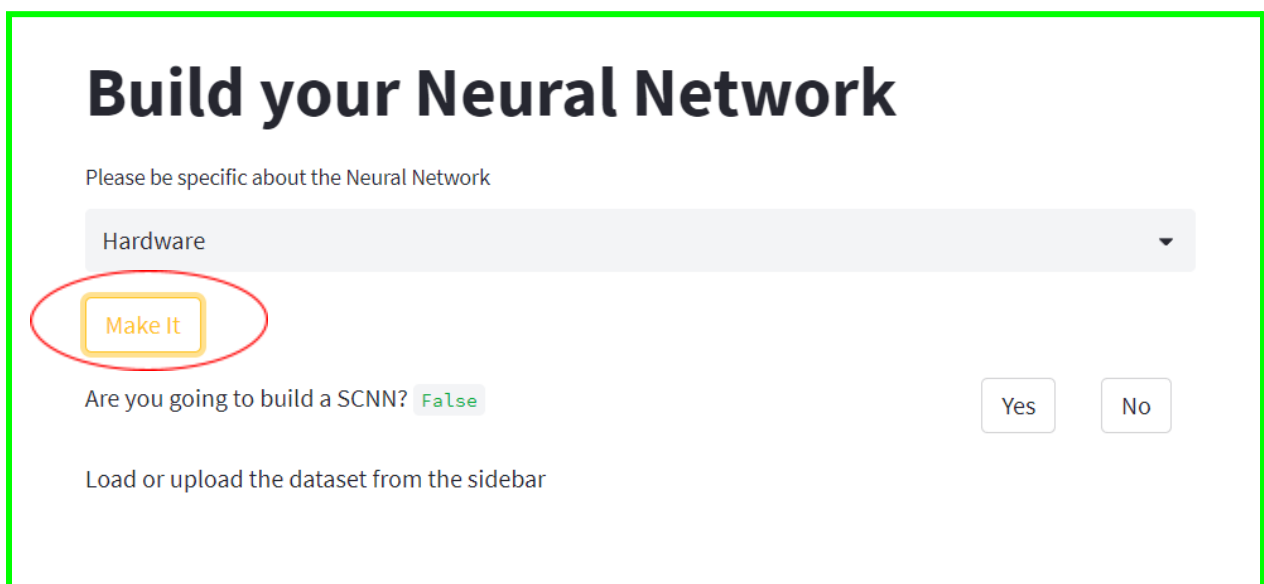
Yes No

Load or upload the dataset from the sidebar

The screenshot shows a web interface titled "Build your Neural Network". Below the title is a subtitle "Please be specific about the Neural Network". There is a dropdown menu currently showing "Hardware". A red arrow points to the dropdown menu, which also lists "Software" as an option. Below the dropdown menu, there is a question "Are you going to build a SCNN?" with a toggle switch set to "False". To the right of this question are two buttons labeled "Yes" and "No". At the bottom of the interface, there is a text label "Load or upload the dataset from the sidebar".

## Step 2

Then, the user has to click on the “Make it” button.



**Build your Neural Network**

Please be specific about the Neural Network

Hardware

Make It

Are you going to build a SCNN? False

Yes No

Load or upload the dataset from the sidebar

The screenshot shows the same web interface as in Step 1. The "Make It" button, which is a yellow button with a black border, is now highlighted with a red oval. The dropdown menu still shows "Hardware". The "Are you going to build a SCNN?" toggle is still set to "False", and the "Yes" and "No" buttons are still present. The text "Load or upload the dataset from the sidebar" remains at the bottom.

### Step 3

If the user is constructing Spiking Neural Network, double-click on “Yes” or else “No”.

## Build your Neural Network

Please be specific about the Neural Network

Hardware

Make It

Are you going to build a SCNN? True

Yes

No

Load or upload the dataset from the sidebar

### Step 4

Load the default dataset or custom dataset. Custom dataset folder should follow the format given below.

Select and Load dataset

mnist

mnist

cifar10

cifar100

Iris

Set the parameters below

units

10

Select and Load dataset

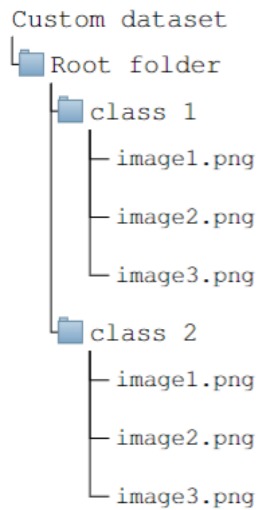
mnist

Load

Upload image dataset

😊 Dataset loaded

The custom dataset folder should follow the format given below.



## Step 5

Choose the **layers** according to the neural network architecture. Give No. of neurons, activation, the name for each layer

Select a layer

Reshape

- Dense
- Conv2D
- DepthwiseConv2D
- MaxPooling2D
- Reshape
- Flatten
- Dropout
- GaussianNoise

Submit

Submitted layers will be displayed in the main page under Added Layers.

Select a layer

Dense

Set the parameters below

units

10

activation

relu

kernel\_initializer

RandomUniform

bias\_initializer

zeros

name

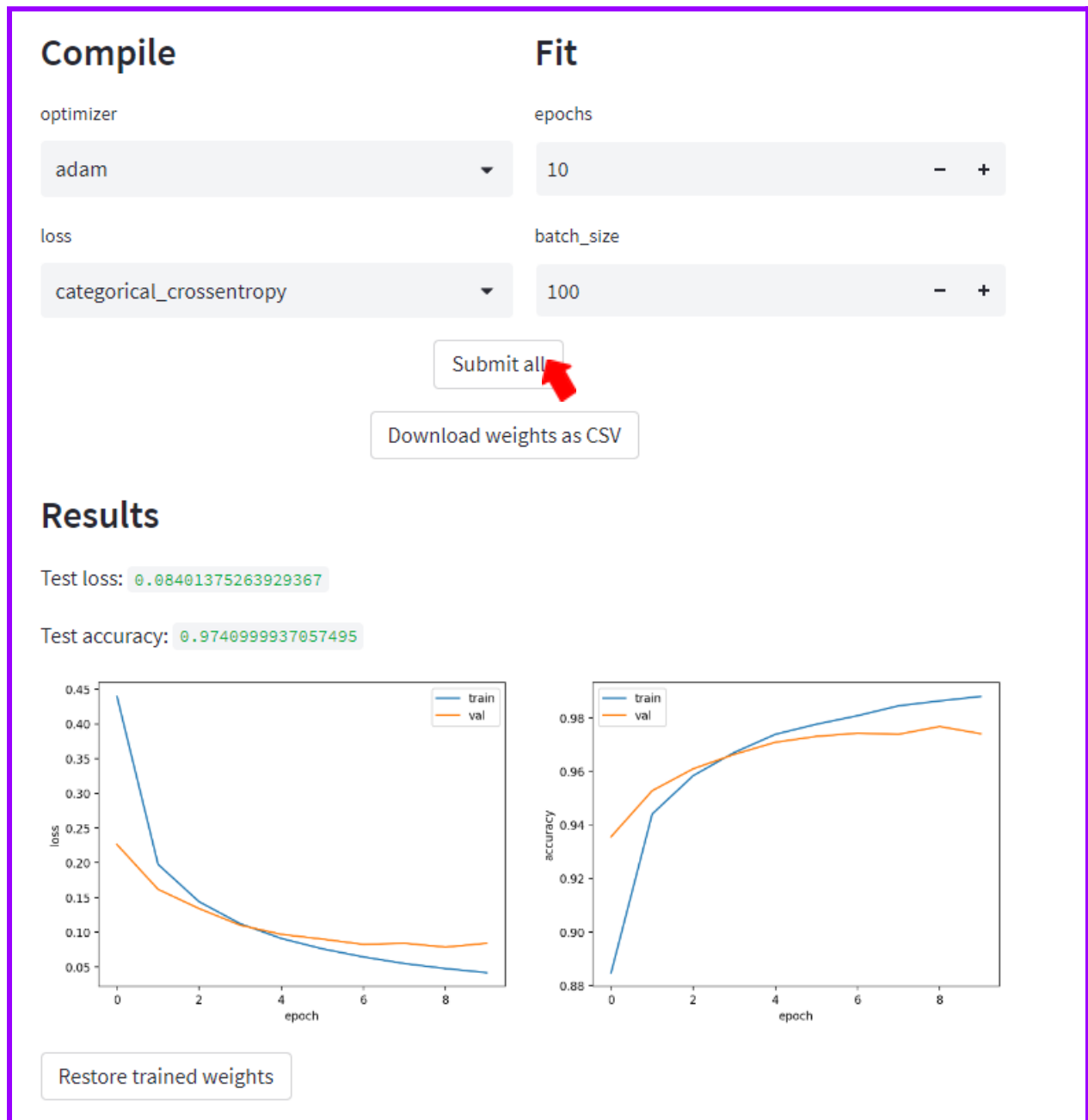
dense\_1

Please update name when same layer is added

Submit

Submitted layers will be displayed in the main page under Added Layers.

Compile and fit the model with suitable parameters then the user can see the result below after clicking the “Submit all” button



## Compile

optimizer

adam

## Fit

epochs

10

loss

categorical\_crossentropy

batch\_size

100

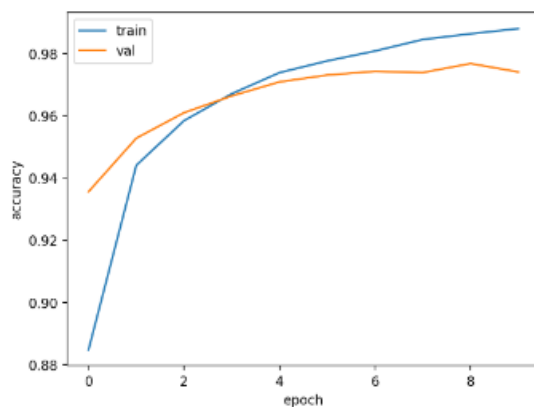
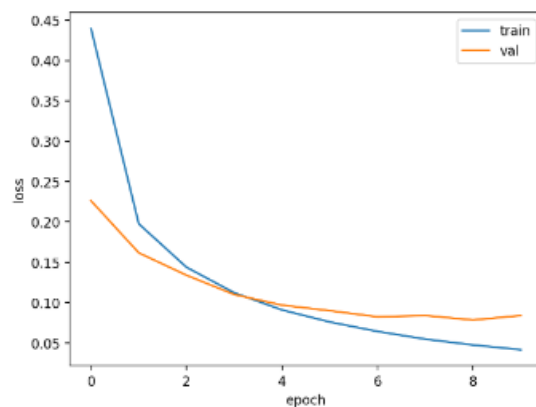
Submit all

Download weights as CSV

## Results

Test loss: 0.08401375263929367

Test accuracy: 0.9740999937057495



Restore trained weights

## **Step 5**

Choose the desired memristor model, then set up the parameters accordingly. Now the user can click on the “Set Up Memristor” button.

Select the memristor

☐ Joglekar  
☒ Prodromakis  
☐ Biolek  
☐ Zha

Enter p value

7

Enter j value

1

Amplitude

1

Frequency

1

☐ Ron-Roff Aging

Set Ron value

100

Set Roff value

16000

Set Rint value

11000

Define the Quatization value here

2

64

Sample Rate

500

Set up Memristor

Now you can see the mapped weights below

Weights for the layer dense\_1 of shape (784, 99)

dense\_1 : Weights

	0	1	2	3	4
0	-0.0335	-0.0177	-0.003	0.0291	0.0253
1	0.0023	0.0111	-0.0408	-0.0197	-0.0385
2	0.0121	-0.0445	-0.0316	0.0122	-0.0069
3	0.0321	0.0437	-0.0287	0.0414	0.0251
4	0.0275	0.0409	0.0169	-0.0212	0.0219
5	0.0428	0.0375	0.0372	-0.0435	-0.0075
6	-0.0366	-0.0198	0.001	-0.0297	0.0089
7	-0.008	0.0297	0.0013	-0.0082	-0.049
8	0.0329	-0.0167	0.0375	-0.0406	0.0409
9	-0.0243	0.043	0.03	-0.0105	-0.0172

dense\_1 : mapped Weights

	0	1	2	3	4
9	-0.071	0.0714	0.0714	-0.071	-0.071
10	0.071	-0.0714	0.0714	0.0714	-0.0707
11	-0.071	0.071	-0.0714	0.0707	-0.0714
12	0.0707	-0.071	0.071	0.071	0.0717
13	-0.0714	-0.072	0.0717	0.0707	0.0714
14	-0.0714	-0.072	0.0714	0.071	-0.072
15	-0.071	0.071	0.0717	0.071	-0.0714
16	-0.071	0.071	0.071	0.0714	0.0714
17	0.071	0.0714	0.071	0.071	-0.0717
18	-0.0717	0.0707	-0.071	-0.0707	-0.0714

dense\_1: Biases

	0
0	-0.0048
1	-0.0162
2	0.0000

dense\_1 : mapped Biases

	0
0	-0.0707
1	-0.071
2	-0.072

Click the “Evaluate” button. Results such as accuracy, loss, precision, Recall of hardware and software neural networks will be shown.

Evaluate							
	Dataset	loss	accuracy	precision	recall	f1 score	Neural network config
0	mnist	0.0840	0.9741	0.9742	0.9741	0.9741	{ "layers": [{ "class_name": "InputLayer", "config": { "dtype": "float32", "name": "input_" }, "activation": "relu", "bias_initializer": { "class_name": "RandomUniform", "config": { "maxval": 0.05, "minval": -0.05 }, "name": "dense1_bias_initializer" }, "class_name": "Dense", "config": { "activation": "relu", "dtype": "float32", "kernel_initializer": { "class_name": "RandomUniform", "config": { "maxval": 0.05, "minval": -0.05 }, "name": "dense1_kernel_initializer" } } } ] }
1	mnist	0.9919	0.7639	0.8373	0.7639	0.7726	{ "layers": [{ "class_name": "InputLayer", "config": { "dtype": "float32", "name": "input_" }, "activation": "relu", "bias_initializer": { "class_name": "RandomUniform", "config": { "maxval": 0.05, "minval": -0.05 }, "name": "dense1_bias_initializer" }, "class_name": "Dense", "config": { "activation": "relu", "dtype": "float32", "kernel_initializer": { "class_name": "RandomUniform", "config": { "maxval": 0.05, "minval": -0.05 }, "name": "dense1_kernel_initializer" } } } ] }