

# run\_neural\_network

This application is designed to operate each neural network API. This application includes APIs, such as CreateNetwork, CreateInputTensor, CreateOutputTensor, LoadNetwork, and RunNetwork.

- Possible Scenario:

Loading and running a neural network to receive processing results from the model is a possible scenario.

## API

| API                 | Return             | Parameter          | Description   |
|---------------------|--------------------|--------------------|---|
| GetOrCreateNetwork  | NeuralNetwork*     | const string& name | Creates a neural network instance when a neural network with the name you entered does not exist. |
| CreateInputTensor   | shared_ptr         | const string &name | Creates an input tensor with a name specifically reserved for the model input                     |
| CreateOutputTensor  | shared_ptr         | const string &name | Creates an output tensor with a name specifically reserved for the model output                   |
| LoadNetwork         | bool               | const string &bin  | Loads a neural network into memory  |
| RunNetwork          | bool               | stat_t &           | Runs the neural network   |
| UnloadNetwork       | void               | -                  | Unloads a neural network from memory  |
| GetInputTensor      | shared_ptr         | const string &name | Gets an input tensor by name  |
| GetInputTensor      | shared_ptr         | size_t index       | Gets an input tensor by index   |
| GetOutputTensor     | shared_ptr         | const string &name | Gets an output tensor by name   |
| GetOutputTensor     | shared_ptr         | size_t index       | Gets an output tensor by index  |
| GetAllInputTensors  | vector<shared_ptr> | -                  | Gets all Input tensor   |
| GetAllOutputTensors | vector<shared_ptr> | -                  | Gets all output tensor  |

| API                  | Return | Parameter | Description                      |
|----------------------|--------|-----------|----------------------------------|
| GetInputTensorCount  | size_t | -         | Gets the count of Input tensor.  |
| GetOutputTensorCount | size_t | -         | Gets the count of Output tensor. |

## Using the Neural Network API

The methods listed below are examples of "google\_net use".

- Create a neural network.

```
auto network = GetOrCreateNetwork("google_net.bin");
```

- Create a Input Tensor.

```
const shared_ptr<Tensor>& input_tensor(network->CreateInputTensor("data_0"));
```

- Create a Output Tensor.

```
const shared_ptr<Tensor>& output_tensor(network->CreateOutputTensor("prob_1"));
```

- Load a neural network into memory

```
bool result;
result = network->LoadNetwork("../res/ai_bin/google_net.bin");
```

- Run the neural network into memory

```
bool result;
stat_t stat = { 0, };
result = network->RunNetwork(stat);
```

- Unload a neural network from memory

```
network->UnloadNetwork();
RemoveNetwork("google_net.bin");
```

- Get an InputTensor using the tensor name.

```
const shared_ptr<Tensor> &input_tensor(network->GetInputTensor("data_0"));
```

- Get an InputTensor using the tensor index.

```
const shared_ptr<Tensor> &input_tensor(network->GetInputTensor(0));
```

- Get an OutputTensor using the tensor name.

```
const shared_ptr<Tensor>& output_tensor(network->GetOutputTensor("prob_1"));
```

- Get an OutputTensor using the tensor index.

```
const shared_ptr<Tensor>& output_tensor(network->GetOutputTensor(0));
```

- Get all InputTensors.

```
const vector<shared_ptr<Tensor>>& all_input_tensor(network->GetAllInputTensors());
```

- Get all OutputTensors.

```
const vector<shared_ptr<Tensor>>& all_output_tensor(network->GetAllOutputTensors());
```

- Get the number of InputTensors.

```
size_t input_tensor_count = network->GetInputTensorCount();
```

- Get the number of OutputTensors.

```
size_t output_tensor_count = network->GetOutputTensorCount();
```

# Running Neural Network

- Click [Start] on the web page.

| No. | Application name  | Status  | Setup   |
|-----|---|---|---|
| 1   | run_neural_network<br>Installed date : 2023-12-04 T 00:00:57<br>Version : 1.0<br><a href="#">Uninstall</a> <a href="#">Go App</a> | Stopped<br><a href="#">Start</a> <a href="#">Health</a> | Priority<br><input checked="" type="radio"/> Low <input type="radio"/> Medium<br><input type="radio"/> High<br>Auto start<br><input type="checkbox"/> Enable<br><a href="#">Apply</a> |

- Once the application is running, click [Go App].

| No. | Application name  | Status   | Setup   |
|-----|---|--|---|
| 1   | run_neural_network<br>Installed date : 2023-12-04 T 00:00:57<br>Version : 1.0<br><a href="#">Uninstall</a> <a href="#">Go App</a> | Running<br><a href="#">Stop</a> <a href="#">Health</a> | Priority<br><input checked="" type="radio"/> Low <input type="radio"/> Medium<br><input type="radio"/> High<br>Auto start<br><input type="checkbox"/> Enable<br><a href="#">Apply</a> |

- To run the neural network, click all [Send] buttons in order from the top. [CreateNetwork]>[CreateInputTensor]>[CreateOutputTensor]>[LoadNetwork]>[RunNetwork].

## Run Neural Network

1. CreateNetwork  
 : Enter the name of the neural network model.  
[Send](#)

2. CreateInputTensor  
 : name of input tensor.  
[Send](#)

3. CreateOutputTensor  
 : name of output tensor.  
[Send](#)

4. LoadNetwork  
[Send](#)

5. RunNetwork  
[Send](#)

### Optional API

- GetInputTensor\_Index  
 : Enter the name of input tensor.  
[Send](#)

- GetOutputTensor\_Index  
 : name of output tensor.  
[Send](#)

- GetInputTensor\_Name  
 : Index of input tensor.  
[Send](#)

- GetOutputTensor\_Name  
 : Index of output tensor.  
[Send](#)

- GetAllInputTensors  
[Send](#)

- GetAllOutputTensors  
[Send](#)

- GetInputTensorCount  
[Send](#)

- GetOutputTensorCount  
[Send](#)

### Unload Network

- UnloadNetwork  
[Send](#)

## Run Neural Network

1. CreateNetwork  
 : Enter the name of the neural network model.  
[Send](#) [Pass](#)

2. CreateInputTensor  
 : name of input tensor.  
[Send](#) [Pass](#)

3. CreateOutputTensor  
 : name of output tensor.  
[Send](#) [Pass](#)

4. LoadNetwork  
[Send](#) [Pass](#)

5. RunNetwork  
[Send](#) [Pass](#)

### Optional API

- GetInputTensor\_Index  
 : Enter the name of input tensor.  
[Send](#)

- GetOutputTensor\_Index  
 : name of output tensor.  
[Send](#)

- GetInputTensor\_Name  
 : Index of input tensor.  
[Send](#)

- GetOutputTensor\_Name  
 : Index of output tensor.  
[Send](#)

- GetAllInputTensors  
[Send](#)

- GetAllOutputTensors  
[Send](#)

- GetInputTensorCount  
[Send](#)

- GetOutputTensorCount  
[Send](#)

### Unload Network

- UnloadNetwork  
[Send](#) [Pass](#)

- To terminate the neural network, click [Send] below [UnloadNetwork].

## Run Neural Network

1. CreateNetwork  
 Enter the name of the neural network model.

2. CreateInputTensor  
 name of input tensor.

3. CreateOutputTensor  
 name of output tensor.

4. LoadNetwork

5. RunNetwork

### Optional API

- GetInputTensor\_Index  
 Enter the name of input tensor.

- GetOutputTensor\_Index  
 name of output tensor.

- GetInputTensor\_Name  
 Index of input tensor.

- GetOutputTensor\_Name  
 Index of output tensor.

- GetAllInputTensors

- GetAllOutputTensors

- GetInputTensorCount

- GetOutputTensorCount

### Unload Network

- UnloadNetwork

## Run Neural Network

1. CreateNetwork  
 Enter the name of the neural network model.

2. CreateInputTensor  
 name of input tensor.

3. CreateOutputTensor  
 name of output tensor.

4. LoadNetwork

5. RunNetwork

### Optional API

- GetInputTensor\_Index  
 Enter the name of input tensor.

- GetOutputTensor\_Index  
 name of output tensor.

- GetInputTensor\_Name  
 Index of input tensor.

- GetOutputTensor\_Name  
 Index of output tensor.

- GetAllInputTensors

- GetAllOutputTensors

- GetInputTensorCount

- GetOutputTensorCount

### Unload Network

- UnloadNetwork

- To use the Optional API, create the network and tensor.

## Run Neural Network

1. CreateNetwork  
 Enter the name of the neural network model.

2. CreateInputTensor  
 name of input tensor.

3. CreateOutputTensor  
 name of output tensor.

4. LoadNetwork

5. RunNetwork

### Optional API

- GetInputTensor\_Index  
 Enter the name of input tensor.

- GetOutputTensor\_Index  
 name of output tensor.

- GetInputTensor\_Name  
 Index of input tensor.

- GetOutputTensor\_Name  
 Index of output tensor.

- GetAllInputTensors

- GetAllOutputTensors

- GetInputTensorCount

- GetOutputTensorCount

### Unload Network

- UnloadNetwork

# Building Application

- Build application.

```
$ APP_NAME=run_neural_network SDK_VER=24.06.14(your SDK version)
SOC=[cv5, jetson] docker compose up
$ docker compose down --remove-orphans
```

2. Check the build results in current directory. If successful, you will be able to find the cap file.