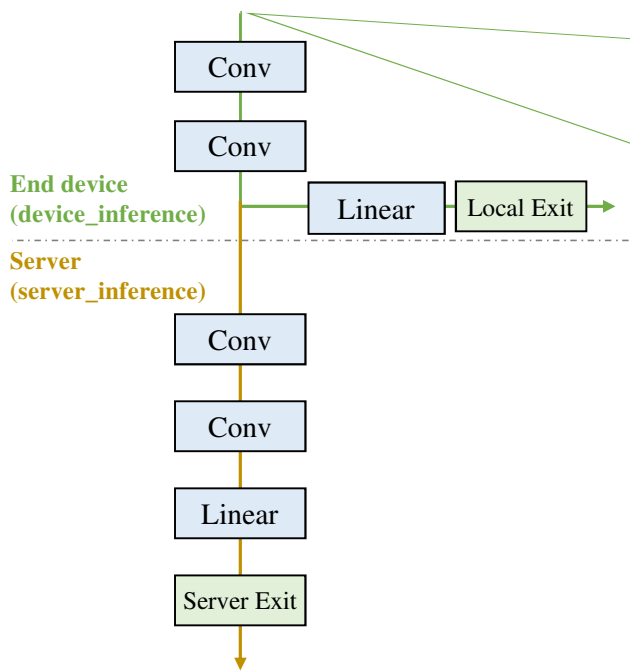


An example of a trained DDNN model



**dnn\_c\_lib.h**

```
// C impl. of the convolution layer.
float*[] conv_layer (float*[] x) {
    y = convolution (x);
    return y;
}
...

// C impl. of the fully connected layer.
float*[] linear_layer (float*[] x) {
    y = linear (x);
    return y;
}
...
```

**dev\_model\_infer.c**

```
#include "dnn_c_lib.h"
#include "tool_lib.h"
#define NUM_DEV_LAYERS      2
#define ENTROPY_THRE       t

float*[] device_inference (float* x[]) {
    y1 = conv_layer (x);
    y2 = conv_layer (y1);
    y3 = linear_layer (y2);
    if (entropy(y3) > ENTROPY_THRE)
        y3 = req_server_inference (y2, NUM_DEV_LAYERS);
    return y3;
}
```

**tool\_lib.h**

```
#include "mqtt.h"
#define SERVER_IP      192.168.1.1

mqtt_obj server;

void platform_init () {
    server = mqtt_connect(SERVER_IP);
}

void platform_final () {
    mqtt_disconnect(server);
}

float entropy (float*[] x) {
    return cal_entropy(x);
}

float*[] req_server_inference (float*[] x, int num_layer ) {
    char[] payload = Convert2Char(x)+
        Convert2Char(num_layer);
    mqtt_publish(server, payload);
    mqtt_recv_subscribed_msg(server, y);
    return y;
}
```