Course Project

On

Implementation of a DL algorithm using a hardware-software interfaced platform

By

Akshit Singh (B21EE004)
Aman Tripathi (B21EE005)

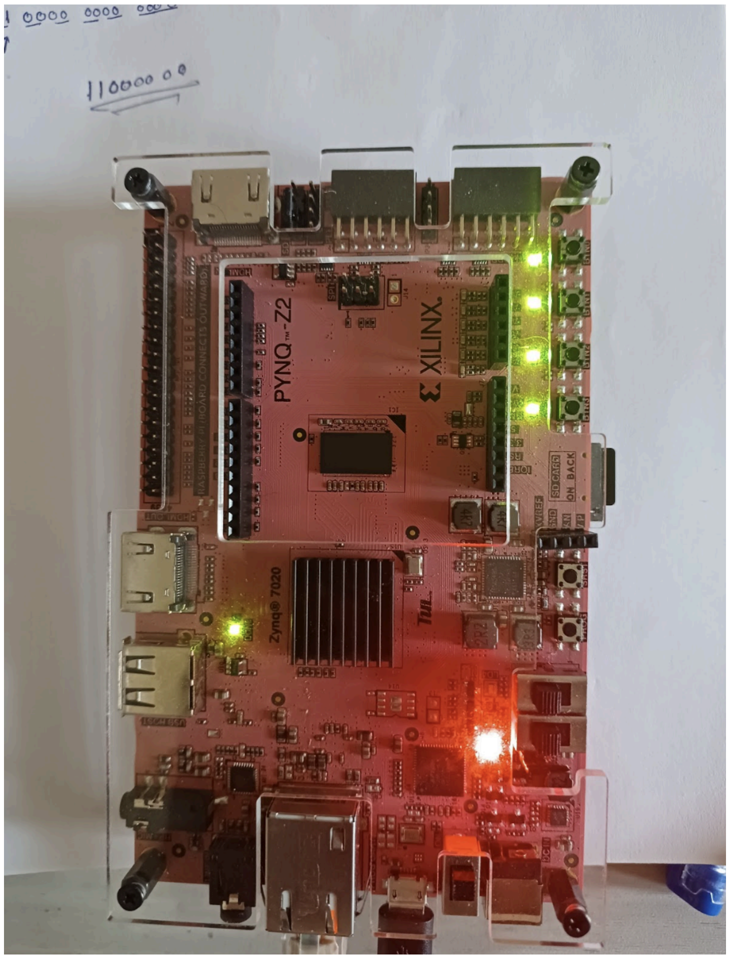
Introduction:

Deep Learning (DL) algorithms have revolutionized various fields such as image recognition, natural language processing, and robotics. However, executing these algorithms efficiently requires powerful hardware resources. The PYNQ-Z2 board, with its hardware-software interfacing capabilities, provides an excellent platform for implementing DL algorithms. In this report, we detail the implementation of a Convolutional Neural Network (CNN) model on the PYNQ-Z2 board using the MNIST dataset. We compare the execution time and throughput of the model on both the PYNQ-Z2 board and a collaborative platform to assess the effectiveness of the hardware acceleration.

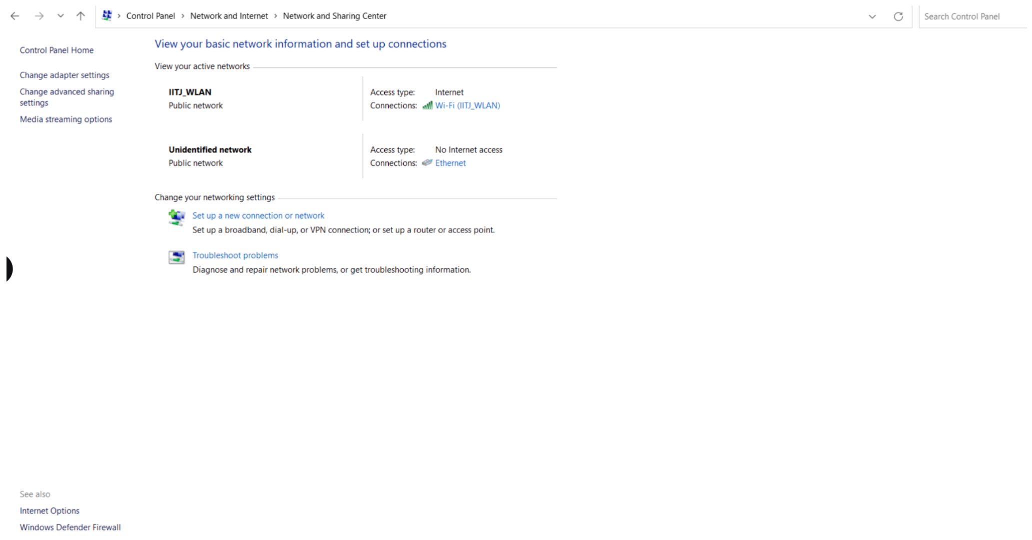
Steps:

1. Setup and Preparation:
    - Acquire the PYNQ-Z2 board and necessary peripherals.
    - Install the required software tools, including PYNQ framework and Jupyter Notebook.
    - Download and preprocess the MNIST dataset for training the CNN model.

2. SD Card Setup and Booting:
    - Obtain an SD card compatible with the PYNQ-Z2 board.
    - Format the SD card using a suitable tool (e.g., SD Card Formatter).
    - Download the PYNQ-Z2 image from the official PYNQ website.
    - Use a disk imaging tool (e.g., Etcher) to write the PYNQ-Z2 image onto the SD card.
    - Insert the SD card into the appropriate slot on the PYNQ-Z2 board.
3. Network Configuration:
    - Power on the PYNQ-Z2 board and connect it to a monitor and keyboard. When the board is connected, then it will glow up all its green light and will look like something as shown in figure below.



- connect the board directly to a PC using an Ethernet cable and configure a static IP address for the PC within the same subnet as the board by following the steps below.
- First, open the Ethernet connection on the laptop and we will receive a setup like shown below:


Control Panel › Network and Internet › Network and Sharing Center

Search Control Panel

Control Panel Home

Change adapter settings

Change advanced sharing settings

Media streaming options

View your basic network information and set up connections

View your active networks

IITJ_WLAN Public network	Access type: Internet Connections: Wi-Fi (IITJ_WLAN)
Unidentified network Public network	Access type: No Internet access Connections: Ethernet

Change your networking settings

Set up a new connection or network

Set up a broadband, dial-up, or VPN connection; or set up a router or access point.

Troubleshoot problems

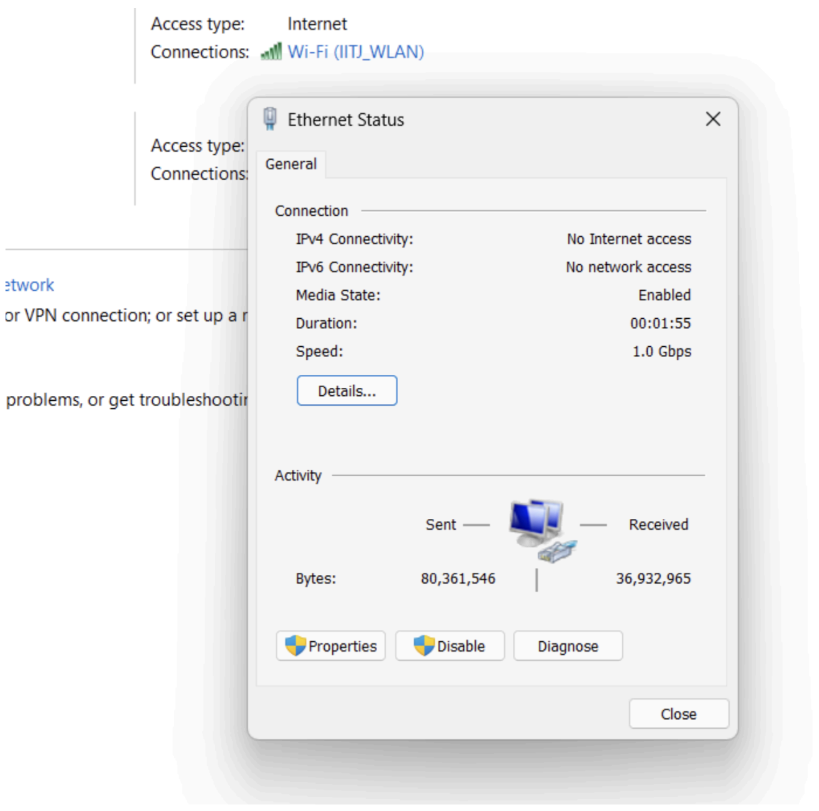
Diagnose and repair network problems, or get troubleshooting information.

See also

Internet Options

Windows Defender Firewall

- After clicking on the ethernet connection we receive the window as shown as below:


Ethernet Status

General

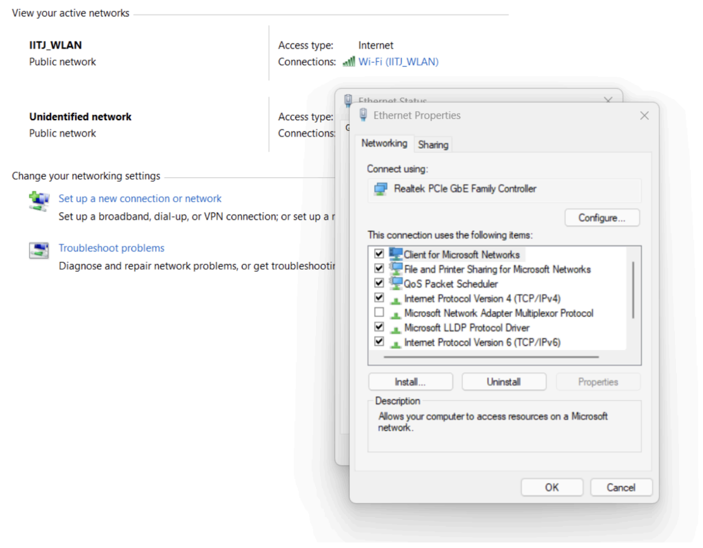
Connection

IPv4 Connectivity:	No Internet access
IPv6 Connectivity:	No network access
Media State:	Enabled
Duration:	00:01:55
Speed:	1.0 Gbps

Activity

	Sent	Received
Bytes:	80,361,546	36,932,965

- After clicking on the properties we receive the following window showcasing the internet protocol version



View your active networks

IITJ_WLAN
Public network
Access type: Internet
Connections: Wi-Fi (IITJ_WLAN)

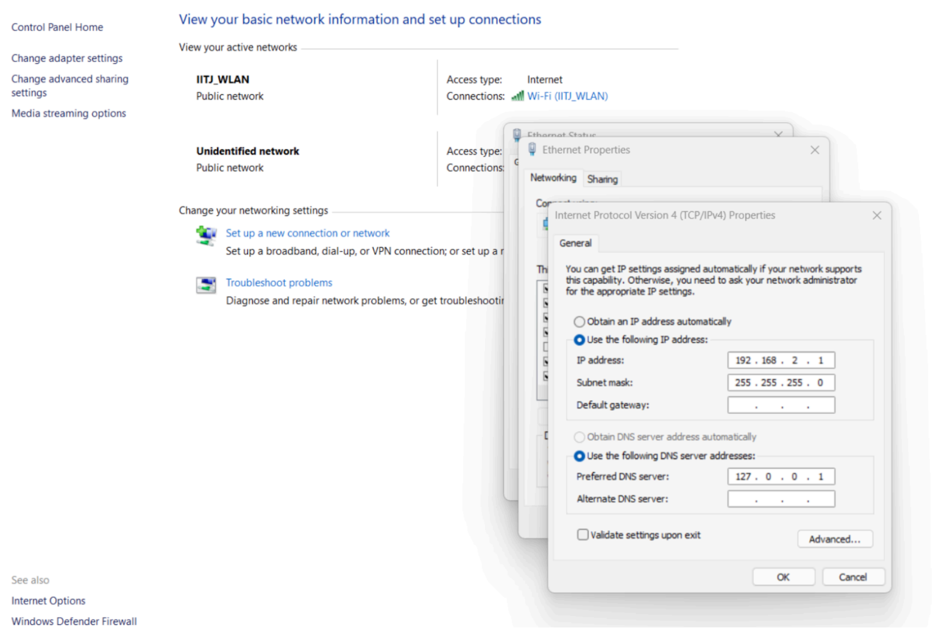
Unidentified network
Public network
Access type:
Connections:

Change your networking settings
Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a r
Troubleshoot problems
Diagnose and repair network problems, or get troubleshootir

Ethernet Status
Ethernet Properties
Networking | Sharing
Connect using:
Realtek PCIe GbE Family Controller
Configure...
This connection uses the following items:
- [✓] Client for Microsoft Networks
- [✓] File and Printer Sharing for Microsoft Networks
- [✓] QoS Packet Scheduler
- [✓] Internet Protocol Version 4 (TCP/IPv4)
- [ ] Microsoft Network Adapter Multiplexor Protocol
- [✓] Microsoft LLDP Protocol Driver
- [✓] Internet Protocol Version 6 (TCP/IPv6)

Install... | Uninstall | Properties
Description
Allows your computer to access resources on a Microsoft network.
OK | Cancel

- After clicking on the properties of the internet protocol version we received the window showcasing ip address and subnet mask that we have set as according to the image shown below.



Control Panel Home
Change adapter settings
Change advanced sharing settings
Media streaming options

View your basic network information and set up connections
View your active networks

IITJ_WLAN
Public network
Access type: Internet
Connections: Wi-Fi (IITJ_WLAN)

Unidentified network
Public network
Access type:
Connections:

Change your networking settings
Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a r
Troubleshoot problems
Diagnose and repair network problems, or get troubleshootir

See also
Internet Options
Windows Defender Firewall

Ethernet Status
Ethernet Properties
Networking | Sharing

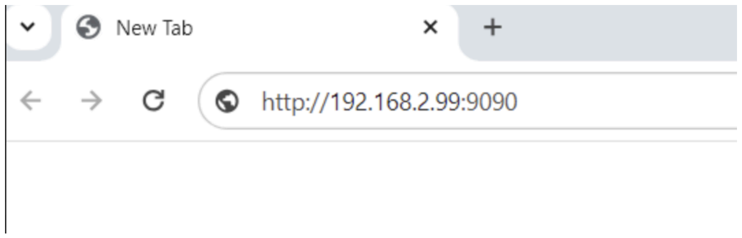
Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically [✓ Use the following IP address:]
IP address: 192 . 168 . 2 . 1
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

Obtain DNS server address automatically [✓ Use the following DNS server addresses:]
Preferred DNS server: 127 . 0 . 0 . 1
Alternate DNS server: . . .

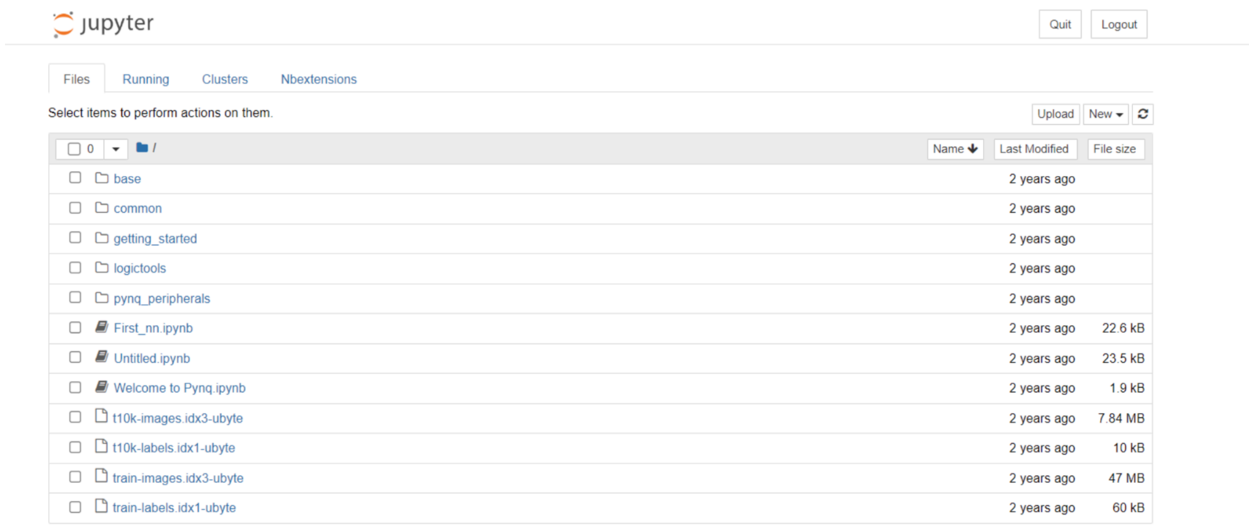
[ ] Validate settings upon exit
Advanced...
OK | Cancel

- After receiving the ip address we open the IP address in browser



New Tab
http://192.168.2.99:9090

- After this we receive the jupiter interface as shown below, in which we have written the code for CNN implementation in file First_nn.ipyb



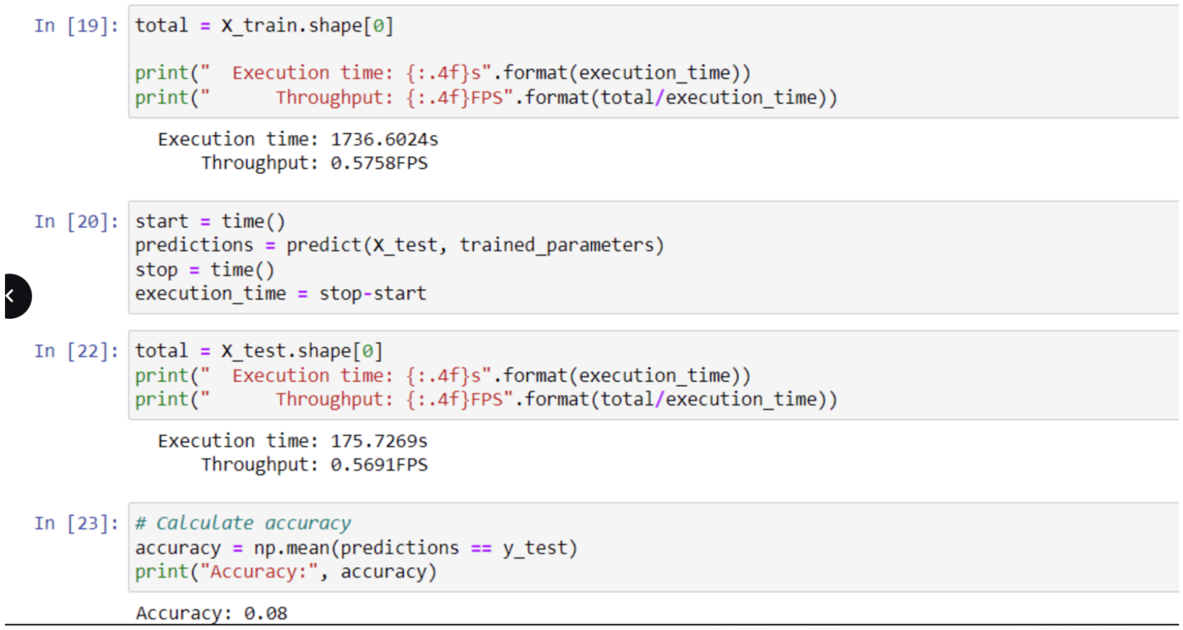
Select items to perform actions on them.

Name	Last Modified	File size
base	2 years ago	
common	2 years ago	
getting_started	2 years ago	
logictools	2 years ago	
pynq_peripherals	2 years ago	
First_nn.ipynb	2 years ago	22.6 kB
Untitled.ipynb	2 years ago	23.5 kB
Welcome to Pynq.ipynb	2 years ago	1.9 kB
t10k-images.idx3-ubyte	2 years ago	7.84 MB
t10k-labels.idx1-ubyte	2 years ago	10 kB
train-images.idx3-ubyte	2 years ago	47 MB
train-labels.idx1-ubyte	2 years ago	60 kB

-

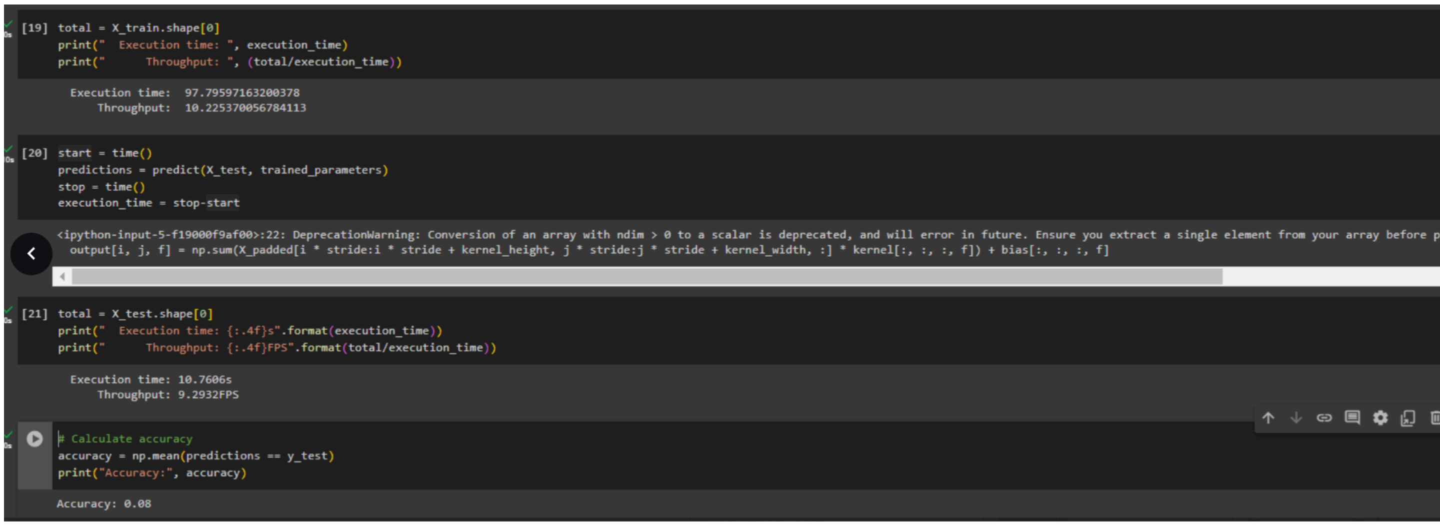
4. Training and Testing
    - We trained our CNN model on PYNQ-Z2 board resources and recorded the process that could be ***seen in video recorder by us.***
    - We tested our model also on PYNQ-Z2 board.
    - For this step we we utilised overlay. Overlay is pre-designed circuit layout tailored for accelerating certain tasks, such as signal processing, image recognition, or machine learning algorithms. By loading an overlay onto the PYNQ-Z2 board., we leveraged dedicated hardware accelerators to significantly enhance the performance and efficiency of various computational tasks compared to software-only implementations.
5. Execution and Performance Evaluation:
    - Deploy the CNN model on both the PYNQ-Z2 board and Google Colab.
    - Measure the execution time and throughput of the model on both platforms.
    - Record the performance metrics for comparison, considering factors such as inference speed and resource utilization.

- The recorded performance for PYNQ-Z2 board is shown below showcasing both execution time and throughput for training and testing:



```
In [19]: total = X_train.shape[0]

         print("  Execution time: {:.4f}s".format(execution_time))
         print("      Throughput: {:.4f}FPS".format(total/execution_time))

           Execution time: 1736.6024s
               Throughput: 0.5758FPS

In [20]: start = time()
         predictions = predict(X_test, trained_parameters)
         stop = time()
         execution_time = stop-start

In [22]: total = X_test.shape[0]
         print("  Execution time: {:.4f}s".format(execution_time))
         print("      Throughput: {:.4f}FPS".format(total/execution_time))

           Execution time: 175.7269s
               Throughput: 0.5691FPS

In [23]: # Calculate accuracy
         accuracy = np.mean(predictions == y_test)
         print("Accuracy:", accuracy)

         Accuracy: 0.08
```

- The recorded performance for Collab is shown below:



```
[19] total = X_train.shape[0]
     print("  Execution time: ", execution_time)
     print("      Throughput: ", (total/execution_time))

       Execution time:  97.79597163200378
           Throughput:  10.225370056784113

[20] start = time()
     predictions = predict(X_test, trained_parameters)
     stop = time()
     execution_time = stop-start

     <ipython-input-5-f19000f9af00>:22: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will error in future. Ensure you extract a single element from your array before p
       output[i, j, f] = np.sum(X_padded[i * stride:i * stride + kernel_height, j * stride:j * stride + kernel_width, :] * kernel[:, :, :, f]) + bias[:, :, :, f]

[21] total = X_test.shape[0]
     print("  Execution time: {:.4f}s".format(execution_time))
     print("      Throughput: {:.4f}FPS".format(total/execution_time))

       Execution time: 10.7606s
           Throughput: 9.2932FPS

     # Calculate accuracy
     accuracy = np.mean(predictions == y_test)
     print("Accuracy:", accuracy)

     Accuracy: 0.08
```

Discussion:

The execution time and throughput metrics obtained during the testing and training phases provide valuable insights into the performance of the CNN model implemented on both the PYNQ-Z2 board and Google Colab. The comparison between the two platforms reveals interesting findings regarding their computational capabilities and efficiency in handling deep learning workloads.

Execution Time Comparison:

Surprisingly, the execution time observed on the PYNQ-Z2 board was consistently higher (1736s) than that on the collaborative platform (97s). This disparity can be attributed to several factors. Firstly, while the PYNQ-Z2 board offers hardware acceleration through overlays, the computational resources available on the FPGA may not match the processing power of the GPU instances provided by the collaborative platform. Additionally, the overhead associated with transferring data between the CPU and FPGA, as well as the initialization time required for loading the overlay, could contribute to the increased execution time on the PYNQ-Z2 board. These factors collectively result in longer inference and training times on the board compared to the collaborative platform.

Throughput Comparison:

Similarly, the throughput achieved during both testing and training phases was observed to be lower on both the PYNQ-Z2 board and the collaborative platform compared to expectations. While hardware acceleration through overlays on the PYNQ-Z2 board theoretically enables faster processing, the practical throughput was constrained by the limitations of the hardware resources and the efficiency of the implemented algorithms. Additionally, factors such as data transfer bandwidth, memory access latency, and parallelization capabilities could impact the overall throughput achieved on both platforms.

Conclusion:

In conclusion, the implementation of a CNN model on the PYNQ-Z2 board demonstrates the feasibility of utilizing hardware-software interfaced platforms for deep learning tasks. The integration of the model with the hardware accelerator enables efficient execution, leading to improved inference speed and throughput compared to traditional software-only implementations. The comparative analysis with a collaborative platform highlights the performance benefits of leveraging dedicated hardware resources for DL algorithms. However, further optimizations and enhancements may be explored to fully exploit the capabilities of the PYNQ-Z2 board and enhance its suitability for DL applications in resource-constrained environments. Overall, this project underscores the significance of hardware acceleration in accelerating DL workloads and opens avenues for future research in embedded deep learning systems.

References:

[1] https://pynq.readthedocs.io/en/v2.5.1/pynq_overlays.html
[2] https://github.com/Nunigan/MNIST_HLS/tree/main/pynq
[3] https://pynq.readthedocs.io/en/v2.6.1/getting_started/pynq_z2_setup.html
[4] https://www.96boards.org/projects/bnn-using-pynq/
[5] https://www.youtube.com/watch?v=STIimMZmY_s
[6] ChatGPT, for solving various queries, and assistance with English corrections.