جامعة الأمير مقرن بن عبد العزيز
University of Prince Mugrin

Department of Artificial Intelligence

College of Computer and Cyber Sciences

Introduction to Deep Learning

Neural Network Design, Training, and Evaluation with PyTorch

1. Learning Objectives

By the end of this lab, students will:

- Define and Implement a **Neural Network** Model.
- **Train**, **Test**, and **Evaluate** the Model.
- Explore Deep Learning **Datasets**.

2. Lab Requirements

Software:

- Python or Anaconda Jupyter notebook.
- PyTorch.
- CUDA.

Hardware:

- Students should use the lab devices for running their deep learning models.

3. Explanation of Key Concepts

- **CPU vs GPU:**

  CPU (Central Processing Unit): The brain of the computer, handling general tasks like running software, browsing, and multitasking. It has a few powerful cores optimized for sequential processing. Good for general computing and logic-based tasks.

  GPU (Graphics Processing Unit): Designed for heavy parallel tasks like gaming, video editing, and AI computations. It has many smaller cores that process multiple tasks simultaneously, making it much faster for specific workloads. Best for graphics, AI, and tasks requiring high-speed parallel processing.

- **MNIST Dataset:**

  The MNIST dataset (Modified National Institute of Standards and Technology dataset) is a large collection of handwritten digits (0-9) used for training and testing machine learning models, especially in image recognition and deep learning.

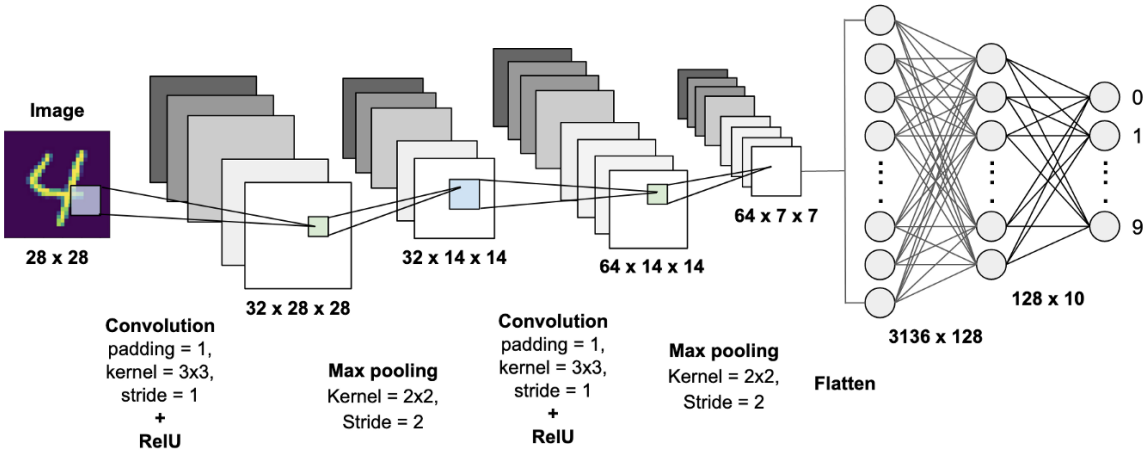
  Key Features of MNIST:

  Size: 70,000 images (60,000 for training + 10,000 for testing).
  Image Format: 28x28 pixels, grayscale (black & white).
  Labels: 10 classes (digits 0 to 9).

4. Activities

➢ Exercise 1: Training on MNIST.

Train a PyTorch model on the Mnist dataset using GPU.

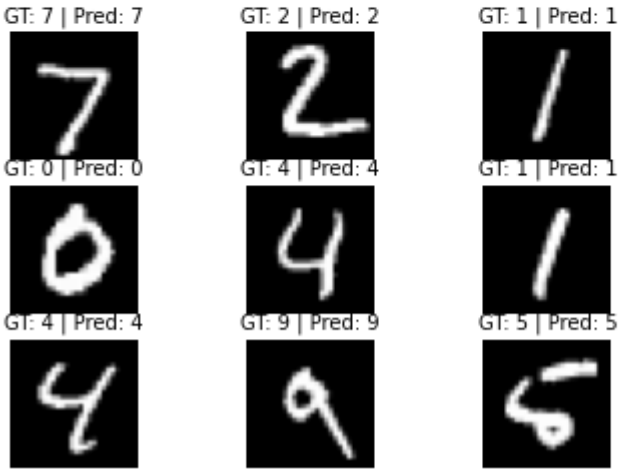



```
# Code provided in the notebook
```

Screenshot of the result:

```
Test set: Average loss: 0.0412, Accuracy: 9863/10000 (99%)

GT #1: 7 | Prediction #1: 7
GT #2: 2 | Prediction #2: 2
GT #3: 1 | Prediction #3: 1
GT #4: 0 | Prediction #4: 0
GT #5: 4 | Prediction #5: 4
GT #6: 1 | Prediction #6: 1
GT #7: 4 | Prediction #7: 4
GT #8: 9 | Prediction #8: 9
GT #9: 5 | Prediction #9: 5
```



5. Tasks

➢ Task 1:

Train the same model provided in Exercise 1 with the CIFAR-10 dataset and add comments to all functions in the notebook:

Add your code here (or just submit your jupyter notebook):

Screenshot of the result:

```
Test set: Average loss: 1.3673, Accuracy: 5067/10000 (51%)

GT #1: 4 | Prediction #1: 4
GT #2: 2 | Prediction #2: 5
GT #3: 1 | Prediction #3: 1
GT #4: 8 | Prediction #4: 8
GT #5: 2 | Prediction #5: 2
GT #6: 8 | Prediction #6: 8
GT #7: 2 | Prediction #7: 2
GT #8: 8 | Prediction #8: 3
GT #9: 1 | Prediction #9: 1
```

GT: 4 | Pred: 4 GT: 2 | Pred: 5 GT: 1 | Pred: 1



GT: 8 | Pred: 8 GT: 2 | Pred: 2 GT: 8 | Pred: 8



GT: 2 | Pred: 2 GT: 8 | Pred: 3 GT: 1 | Pred: 1



6. References

CPU vs. GPU: What's the Difference?
MNIST - Ultralytics YOLO Docs
examples/mnist/main.py at main · pytorch/examples
CIFAR-10 and CIFAR-100 datasets