Practical 3

Aim :-

Convolutional Neural Networks:
Use MNIST fashion dataset and create a classifier to classify fashioning clothing into categories

Objectives :-

i) To implement different deep learning models.
ii) To understand hardware acceleration.
iii) To illustrate concepts of Artificial Intelligence/ Machine learning (AI/ML)

Requirements :-

64 bit windows O.S., python, python libraries - Tensorflow, Numpy, Pandas, Jupyter notebook.

Theory :-

Convolutional Neural Network (CNN) :-

It is a type of a neural network commonly used in deep learning for image recognition, classification and segmentation tasks. The architecture of a CNN is designed to hierarchical of features from input images. In a CNN, the input image is first passed through a series of convolutional

layers each of which applies a set of filters to input image to extract features at different spatial scales. These features are then passed through a pooling layer, which reduces the dimensionality of features while preserving their important spatial information.

Finally, the output of the last pooling layer is passed through one or more fully connected layers, which perform classification or regression based on learned features.

CNN have been very successful in a wide range of computer vision tasks, including image classification, object detection and semantic segmentation.

CNN has wide range of applications:-

1) Image classification.
2) Object detection.
3) Semantic segmentation.
4) Natural language processing.

MNIST Dataset:-
It is a fashion dataset & is a collection of 70,000 grayscale images of 28 x 28 pixels, representing 10 different categories of clothings & accessories where each category

category containing 7,000 images.
They are -

- T-shirt / tops.
- Trousers.
- Pullovers
- Presses
- Coats.

- Sandals.
- Shirts.
- Sneakers.
- Bags
- Ankle boots.

General steps to perform CNN :-

1) Import all the necessary libraries.
2) Load dataset using keras built-in function keras. dataset. Fashion_mnist_load_data().
3) Pre-process data by normalizing the pixel value between 0 & 1, reshaping the images to be of size (28, 28, 1).
4) Define CNN architecture, including the number size of filters, activation functions & pooling layers.
5) Compile the model by specifying the loss function optimizer & evaluation metrics.
6) Train the CNN on training set using the fit() function.
7) Evaluate performance of model.
8) Use the train model to make predictions on new images, if desired, using the predict() function.

Conclusion:-

In this way, we can classify fashion clothing into categories using CNN.