Experiment No 3
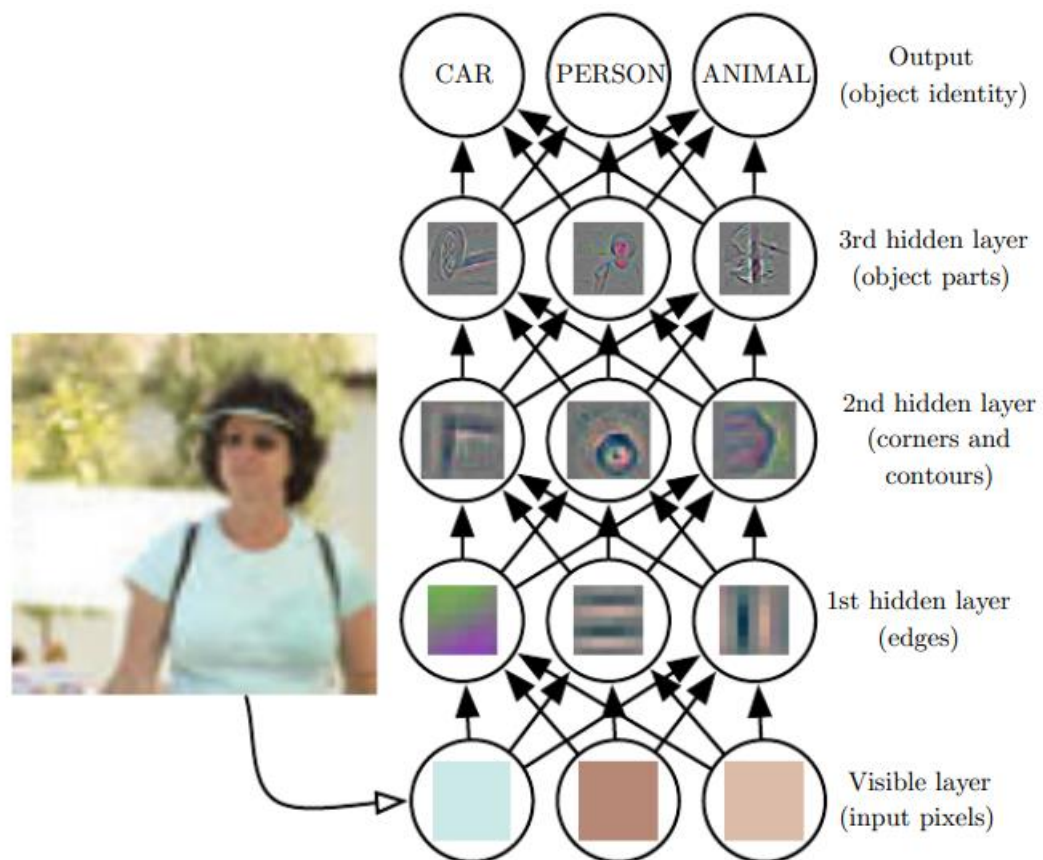Design and develop a deep feed forward neural network to classify fashion products from Fashion MNIST dataset.

Objective:
At the end of this practical session, student will be able to develop a deep neural network using Keras after designing a neural network for digit classification.

Theory:

A. Introduction

With classical ML approaches, it can be very difficult to extract high-level, abstract features from raw data due to variation in different factors. For example, speaker's accent in audio data, position of cat in images. Deep learning solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts with the help of hierarchical layered representations as shown in following figure. The figure shows how a deep learning system can represent the concept of an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges.

In deep neural network, the input is presented at the visible layer, so named because it contains the variables that we are able to observe. Then a series of hidden layers extracts increasingly abstract features from the input data. These layers are called "hidden" because their values are not given in the image; instead the model must determine which concepts are useful for explaining the relationships in the observed data. The images here are visualizations of the kind of feature represented by each hidden unit. Given the pixels of images, the first layer can easily identify edges, by comparing the brightness of neighboring pixels. Given the first hidden layer's description of the edges, the second hidden layer can easily search for corners and extended contours, which are recognizable as collections of edges. Given the second hidden layer's description of the image in terms of corners and contours, the third hidden layer can detect entire parts of specific objects, by finding specific collections of contours and corners. Finally, this description of the image in terms of the object parts it contains can be used to recognize the objects present in the image.

B. Learning in Deep Learning Models:

Learning means finding a set of values for the weights of all layers in a network, such that the network will correctly map example inputs to their associated targets. To control the output of a neural network, you need to be able to measure how far this output is from what you expected. This is the job of the loss function of the network, also called the objective function. The fundamental trick in deep learning is to use this score as a feedback signal to adjust the value of the weights a little, in a direction that will lower the loss score for the current example. The working of deep learning model is summarized in following figure.
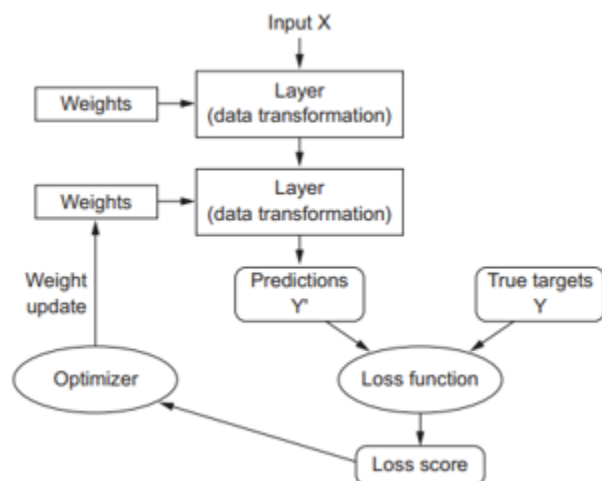
C. Building Deep Learning model in Keras

The quintessential example of a deep learning model is the feedforward deep network or multilayer perceptron (MLP). A multilayer perceptron is just a mathematical function mapping some set of input values to output values.

Models in Keras are defined as a sequence of layers. Fully connected layers in sequence of layers are defined using the Dense class. We can specify the number of neurons

or nodes in the layer as the first argument, and specify the activation function using the activation argument. Likewise we can put many dense layers one after another. These layers are fully connected with previous layer and next layer. First layer in model needs to have input shape to represent the shape of input. This input is transformed from original representation to new representation for predicting output correctly.

Keyword:
Deep Neural Network, ML, Keras

Procedure:
1. Design a deep neural network for classifying fashion product's MNIST dataset.
2. Load MNIST dataset and get training and testing images.
3. Create a model that you designed for mapping inputs to target output.
4. Configure the learning process by choosing a loss function, an optimizer, and some metrics to monitor.
5. Iterate on your training data by calling the fit() method of your model.
6. Evaluate model on testing data to check model's accuracy.