# Artificial Intelligence and Expert

**Homework Assignment III**

**Faculty of Mechanical Engineering**

**Instructor:**

**Dr Esmail Najafi**

**Reza Behbahani Nejad**

**Due date: 1401/03/01**

Human activity recognition (HAR) refers to the task of automatically identifying and classifying human activities based on sensor data collected from wearable devices or other sensors. The goal of HAR is to build machine learning models that can accurately recognize different activities performed by humans, such as walking, running, sitting, standing, or even more complex activities such as climbing stairs or dancing. HAR has important applications in various fields, including healthcare, sports, and smart homes. By using HAR technology, healthcare providers can monitor the physical activity of patients and assess their recovery progress. Sports professionals can use HAR to track athletes' performance and prevent injuries. In smart homes, HAR can be used to control home appliances and adapt the environment to the user's activities and preferences.

Figure 1. Different human activities

The dataset was obtained using the integrated sensors of a smartphone, specifically the Accelerometer, Magnetometer, and Gyroscope. From these sensors, four types of parameter readings were recorded in the X, Y, and Z directions. The X_acc parameter represents acceleration in the X direction, X_AV denotes angular velocity in the X direction, X_mf indicates the magnetic field in the X direction, and X_orien represents the orientation reading in the X direction. Similar parameters were recorded for the Y and Z directions.

The dataset consists of nine distinct classes, with each class represented by a single file. To address class imbalance issues, each file contains 403,605 data points. Data collection was conducted at a sampling frequency of 100 Hz, with a duration of approximately 1.12 hours per class. These details of the dataset are crucial for researchers and practitioners who seek to work with the data and require a clear understanding of its characteristic.

## Experiment I: Multi Layer Perceptron (MLP)

I. What would be the output of a Multi Layer Perceptron (MLP) if only linear activation functions are used, and what problems can it cause? Explain the impact of activation functions on the performance of an MLP network.

II. Plot and explain Leaky-Relu, PReLU, RReLU, ELU, and SELU activation functions.

III. Briefly explain weight initialization and compare three methods of weight initialization: a) assigning equal weights zero to all neurons, b) sampling Gaussian function, and c) Xavier initialization.

IV. For a simple regression task, a simple MLP network is used in Figure 2 with linear activation functions for hidden layers and the output layer. Using MSE loss function, calculate the partial derivatives of the loss function with respect to the biases and weights of the network. Train the network for one epoch and update the network weights and biases. You should write this without using any code. (Hint: Consider the biases equal to zero at first, X1 =3, X2=2, y=5)

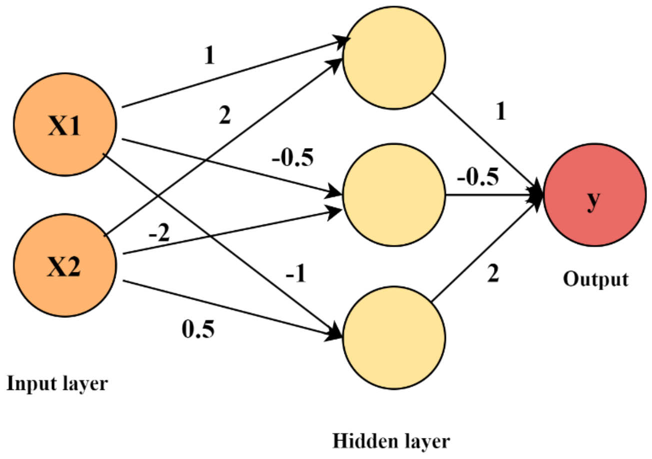

Figure 2: The proposed MLP network

V. Explain what dropout is? What are the differences in its implementation during the training and testing stages of a deep neural network.

VI. Define and explain the concept of batch normalization in the context of machine learning and neural networks.

VII. Consider the Dataset provided and Implement a network with at least 4 hidden layers, and then train the network again using dense layers followed by dropout and batch normalization.

VIII. Compare the results and write your own conclusion. Does the network become overfit, underfit, or train normally? Give reasons for your answer.

IX. What are the popular ways to prevent an MLP from becoming overfit? Mention two ways.

X. What are the hyperparameters of your networks. What are some ways to tune these hyperparameters efficiently? Mention some popular ways such as mesh grid search and random search.

## Experiment II: Recurrent Neural Network (RNN).

I. Explain the basic architecture of an RNN and how it is different from a feedforward neural network. What are the advantages and disadvantages of using an RNN?

II. What is the problem of vanishing and exploding gradients in RNNs? How can this problem be mitigated?

III. Explain the difference between simple RNNs, LSTM, and GRU. In what situations might each type of RNN be most appropriate?

IV. Explain how RNNs are trained using backpropagation through time (BPTT). What are some of the challenges associated with training RNNs, and how can these challenges be addressed?

V. What happens to the gradient if you backpropagate through a long sequence?

VI. Train a network using simple RNN cells

VII. Train a network using LSTM Cells or GRU (you can use both)

VIII. Train a network using bidirectional LSTM or GRU ( you can use both)

IX. Tabular the results, compare them and write your findings.

X. **Bonus**: There's an intriguing paper that presents the Forward-Forward algorithm. This algorithm replaces the forward and backward passes of backpropagation with two forward passes. The first pass involves positive (i.e., real) data, while the second pass involves negative data that can be generated by the network itself. Explore the intriguing aspects of this algorithm by delving into the paper (bearing in mind that it was published in 2022, after Chatgpt knowledge cutoff date of 2021).