Assignment 3: Multilayer Perceptron for Object Classification
Total marks: 100
Due date: 21st of Oct, 2023. (Midnight)

For this assignment you need to implement an MLP architecture and train it to be able to classify 10 different clothing items in fashion_mnist dataset. You should be using two main libraries TensorFlow and Keras for this.

a) These are the steps need to be completed for this part.

1. Load keras.datasets.fashion_mnist data into (train_images, train_labels), (test_images, test_labels).
2. Implement an MLP architecture that can be trained on the fashion dataset. Select the number of layers and number of nodes in each layer according to your consideration. The input and output layers' nodes need to be determined based on the dataset. Please make sure that your network architecture is DIFFERENT from the architecture used in the sample code provided with this assignment.
3. Train the network on the training samples. Your program should print the training accuracy for each epoch. Remember 'accuracy' on a data set is defined by the percentage of correct detection out of total number of samples.
4. Validate the trained model using the test data. Report the final test accuracy. No marks will be deducted if your test accuracy is slightly lower than the best test accuracy, but it should be at least 70%. Please remember that the sample code provided does not calculate the accuracy on the test data set. This part is left for the students to implement. **20**

b) Hyperparameter tuning is a technique to select the (almost) best model architecture for a machine learning problem. In this case, we will try to select the best combination for 3 parameters which are the number of hidden layers, the number of nodes in each hidden layer, and the activation function. For our case, we will select between two different choices for each hyperparameter and try all combinations of hyperparameter values. Finally, we will select the hyperparameter combination that returns the best validation accuracy. You need to complete the below tasks.

1. Consider that your model has three hyperparameters such as the number of hidden layers, the number of nodes in each hidden layer, and the activation function. Each of the hyperparameters can take two values which are (3, 4), (64, 128), ('sigmoid', 'relu') respectively. Implement the grid search algorithm discussed in the class to select each possible hyperparameter combination at a time, define an MLP model with the selected hyperparameter combination, and train the model. The mentioned steps need to be performed in a loop to iterate over each possible hyperparameter combination (in this case the number of possible combinations is 8). Remember to split the train_images into training and validation sets so that you can compute validation accuracy (no cross-validation is needed for this assignment) for each iteration of the loop. Your program must print each combination of hyperparameters and its validation accuracy. Marks will be deducted if this information is not displayed. Selection of hyperparameter combinations, generating the MLP architecture, training the network, and reporting validation accuracy should be done programmatically using loops. Do not use hardcoded hyperparameter combinations and hardcoded network models. **40**
2. Select the hyperparameter combination with the best validation accuracy, and train the MLP model (with the best hyperparameters) using the full training dataset (train_images). Report the test accuracy calculated on the final trained model. Please check the meaning of 'accuracy' defined in part a) of the assignment. **20**
3. Plot the last training and validation accuracies for each hyperparameter combination. This means when you train an MLP corresponding to a hyperparameter combination, save the final training and validation accuracy. Eight different hyperparameter combinations will give you 8 pairs of final training-validation accuracy. Now if you assign indices 1-8 to all hyperparameter combinations (the order of indices doesn't matter here), you can generate a 2D plot such that hyperparameter combination indices can be plotted along the 'x' axis and their corresponding final training accuracy can be plotted along the 'y' axis. Similarly, a plot for hyperparameter combination indices vs. validation accuracies can be generated. Do you find any similarity in patterns between the two plots? Theoretically, are you supposed to see any similarity between these plots? Please describe in detail. **20**

Deliverable:

- Implement the solution on Google Colab platform and upload **BOTH** the .py and .ipynb files on Moodle. All the relevant information should be programmatically printed.

Grading Criteria

1. Your code should run without an error. If it doesn't run, or part of the code doesn't run you will loose 30% of marks (for that part). For example, if part b) of your code doesn't run, you can receive maximum 49 marks (instead of 70) on part b).
2. Late submission: 10% of the awarded marks will be deducted if you are late by one day. 20% for two days. Assignment submission will not be considered if you are more than two days late.