**INM702**

**Programming and Mathematics for Artificial Intelligence**

**Coursework Report**

*By*

Vrushang Bawne (Student Number: 210015700)



**MSc Artificial Intelligence 2021-22**

**[Term 1]**

Contents

[TASK1 2](#_Toc94651894)

[Implementation of game 3](#_Toc94651895)

[TASK2 4](#_Toc94651896)

[Outliers 5](#_Toc94651897)

[Normalization of covariates 7](#_Toc94651898)

[TASK3 9](#_Toc94651900)

[Implementation an artificial neural network 10](#_Toc94651901)

[TASK4 11](#_Toc94651902)

[Dataset-Cifar-10 11](#_Toc94651903)

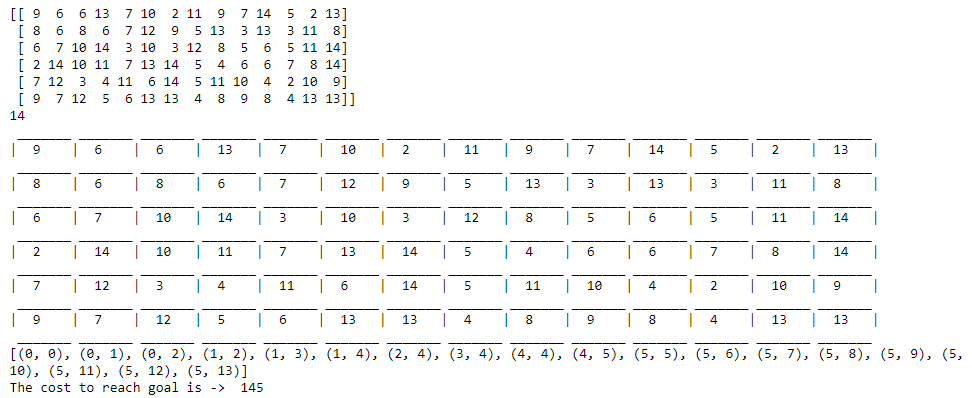
[Model Implementation 11](#_Toc94651904)

[REFLECTION 11](#_Toc94651905)

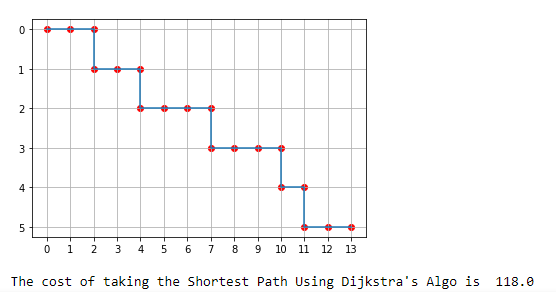
[REFERENCES 12](#_Toc94651906)

# Task 1

The gride consist of a grid populated with random numbers in a uniform distribution. The grid has the number of columns random with the number of rows and this can be chosen by the player.



Implement Dijkstra's algorithm The Dijkstra’s algorithm which will find the shortest path was implemented on o 3X3 grid for both game modes. The path is different every time when the program is running because the distribution of the numbers is different as well.



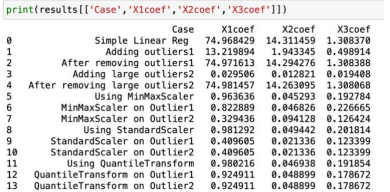
# Task 2

Outliers are present almost everywhere. They can be important in some cases, but for the most part, they tend to disrupt the calculations (with respect to data for modelling). outliers tries to pull the regression line towards itself and thus result in the change of regression coefficient and regression intercept. Change in the coefficient depends upon number of outliers, distance of outlier from the regression line and position. As the number and distance increase the change may reflect with greater magnitude, it depends upon the position of outlier too how sharply the magnitude of coefficient will change. In this report we will investigate different concepts of outliers and different methods to detect and remove them.For the data, have selected to generate three features X1, X2 and X3 with X1 and X2 having mean and standard deviation randomly generated between 0 to 10 and for X3 between 0 to 100. The size of the dataset is also randomly generated. The target variable (y) has been generated as followsy = aX1 + bX2 + cX3 + noise; where a, b, c is randomly generated between different ranges and so is the noise.

Outliers are added for small magnitude and large magnitude to analyze the co-efficient and intercepts. Number of outliers to be added are generated randomly and to different columns

IQR is estimated as the difference between the 75th and the 25th percentiles. K value which we have taken here is 1.5. Any values outside the range of (Q3 + 1.5\*IQR) and (Q1 – 1.5\*IQR) have been considered as outliers and have been removed. The data was then processed by the model, which managed to yield good scores

Normalization of Covariates We have tried three different scalers for normalizing our data, MinMax, Standard and Quantile Transform scaler. We have applied them in three cases each and then checked model performance.

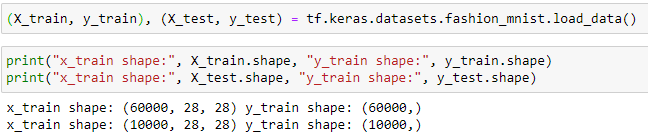


Applied linear regression in different scenarios and tabulated the performance of the model. The detailed output is available in the code. We can see here that the outliers and their magnitude do have a significant effect on the R2 score. However, Quantile Transform can get a high R2 score even with outliers, showing that it can handle the outliers better than the other normalizing methods.

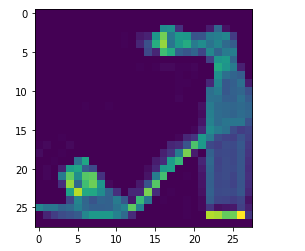
# 

# Task 3

In this task I need to implement an Artificial Neural Network model with a Fashion Mnist data set with articles images. The dataset is form of 784 columns for 28X28 pixels of the images plus 1 column for the labels and 70 000 rows from which 10000are the testing set. First thing that I have made was to transpose the database matrix so each article to be in a row. The first neural network model that I implemented was a model with three layers: one input layer with 784 neurons the number of pixels in the data set, one hidden layer and one output layer both with 10 neurons which is the number of labels and I experimented one by one for the hidden layer with ReLU activation function, then with Sigmoid



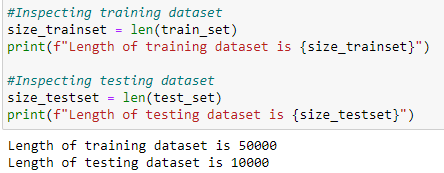
The data used in this report is available as Fashion-MNIST (https://www.kaggle.com/zalandoresearch/fashionmnist) and originally taken from Zalando’s websites. The Fashion-MNIST dataset consists of 70000 28x28 colour images in 10 classes, each of which has 6000 images. Each class is mutually exclusive. There are 60000 training images and 10000 test images. Each image is of height and width of 28 pixels 2D tensors, containing an image tensor and a label. Having a 28x28 px grayscale image of 10 distinct types of classes of clothing where 0: T-shirt/top, 1: Trouser, 2: Pullover, 3: Dress, 4: Coat, 5: Sandal, 6: Shirt, 7: Sneaker, 8: Bag, 9: Ankle boot. Dataset is loaded using TensorFlow library tf. keras.datasets. This dataset class is divided into validation set, testing set and training set for the model. 10,000 images from the training set are devoted for validation, 10,000 are for testing and 50,000 are devoted to training the model



managed to use softmax as the output layer activation function. SO, while calculating the loss, we tried to implement categorical cross entropy loss. But the problem here was that we were not able to resolve the derivatives for the softmax function and the categorical cross entropy loss. This brought us back to the normal loss function as the difference between the prediction and actual values.

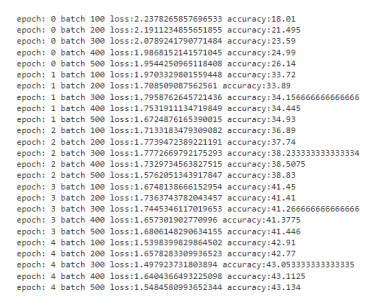
# Task 4

In this Task I had to implement a neural network using PyTorch with Ciffar-10 dataset. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, each of which has 6000 images. contains 60000 RGB images of 32x32 pixels in 10 different classes, with 6000 images per class.



# Implementation of Artificial Neural Network

Artificial neural networks (models) are computational systems made up of multiple layers of interconnected units that are used in deep learning. A neural network can learn how to approximate the computations needed to convert inputs into outputs by sending data through these interconnected units. We started by building an Artificial Neural Network having 3 layers of neurons i.e. the input layer, the hidden layer, and the output layer that generates predictions. There was a need to flatten the image pixels of 3x32x32 pixels equal to 3072, which will act as an initial input to the input layer. Then we set the size of two hidden layers and 120 and 84 neurons, and size of output layer as 10 neurons.

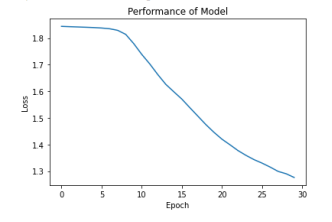


The accuracy of 43.134%.



# CNN (Convolutional Neural Network)

Convolutional Neural Network(CNN) which gave better accuracy results and predicted more number of image correctly compared to the previous ANN model. Conv2d is implemented in CNN. We used 6 layers of Conv2d layers sandwiching between ReLu Layers. As the images are colored and in RGB format we used 3 as our first parameter and 32 as initial input. Kernel size is set to 3 which takes care of filter, max-pooling operation of size 2 \* 2 after two Conv2d layers and padding is used to provide more accurate analysis of the model.



Implemented CNN model and found out that after certain hyper parameter tuning, we managed to get an accuracy of 74%. To conclude, CNN model takes longer time to execute as it uses convolution and pooling layers but gives much better result that an ANN model.

# Reference

MK Gurucharan (2021) Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network. Available at: https://www.upgrad.com/blog/basic-cnn-architecture/ Soumith Chintala (2021) DEEP LEARNING WITH PYTORCH: A 60 MINUTE BLITZ Available at: https://pytorch.org/tutorials/beginner/deep\_learning\_60min\_blitz.html Shreekanya Kotade (2020). Classifying images of everyday objects using a neural network. Available at: https://jovian.ai/shreekanya/03-cifar10-feedforward/v/24?utm\_source=embed Strahinja Zivkovic (2021). Artificial Neural Network with Perceptron on CIFAR10 using PyTorch. Available at: https://datahacker.rs/009-pytorch-building-an-artificial-neural-network-withperceprton-on-cifar10-using-pytorch