INDEX

SL.NO.	NAME OF THE EXPERIMENT	DATE	PAGE NO.	INITIAL OF TEACHER
1.	Outlier Detection	30/9/23	1	
2.	Classification	30/9/23	3	
3.	Image Classification	30/9/23	4	
4.	Image Classification using CIFAR-10	30/10/23	5	
5.	Weight Initialization	30/10/23	7	
6.	Optimiser Analysis	6/10/23	8	
7.	VGGnet-19	13/10/23	9	
8.	RNN for Imdb	20/10/23	10	
9.	LSTM and GRU for imdb	20/11/23	11	
10.	ANN based Image Classifier	27/11/23	12	

Exp. No: 1 Date 30/09/23 Page No: 1

Experiment - 1

OUTLIER DETECTION

Aim: To detect outliers in given data

(a) Using BOXPLOT

Algorithm

1. Import matplotlib library
2. Use Iris Dataset
3. Create a boxplot with the data
4. Display the plot

(b) Using Z-score

1. Import numpy library
2. Call Function detect_outlier_zscore()

//within Function detect_outlier_zscore()
1. SET threshold value
2. SET mean of the data
3. For each sample in data do
    3.1 compute z_score = (i-mean)/std
    3.2 if z_score > threshold THEN
        3.2.1 ADD to outliers
    3.3 END IF
4. END FOR

(c) Using IQR

1. Get data as list
2. Sort the data in ascending order
3. Compute 1st and 3rd quantities of the data.
4. Compute $iqr = q_3 - q_1$
5. calculate lower bound = $q1 - 1.5 * iqr$ and
   upper bound = $q_3 + 1.5 * iqr$
6. FOR each sample in data Do
   6.1 if sample > upper or sample < lower Then
      6.1.1 ADD to outliers
   6.2 END IF
7. END FOR

Result: Program executed and o/p obtained

Exp. No: 2 Date 30/09/23 Page No: 3

Experiment - 2

Aim: Implement linear classification on iris dataset

Algorithm

1. Import necessary libraries
2. Load iris dataset
3. split the dataset x_train, y_train, x_pred, y_pred
4. classifier & import svm
5. bayes & import multinomial NB
6. lr & import linear regression
7. Fit the dataset for (x_train, y_train) for all 3 classifier
8. Using accuracy score from sklearn.metric calculate accuracy of prediction

Result:

Program executed successfully.

17/11/23

Exp. No: 3 Date 30/9/23 Page No: 4

Experiment-3

Aim: Implement a neural network to classify image for a prebuilt dataset

Algorithm

1. Start
2. Import necessary libraries
3. load dataset (mnist dataset)
4. Normalize image values to range (0,1)
5. Create a sequential neural network model using keras model sequential
   - 5.1 Add Flatten layer to convert to 28×28 images to 1D array
   - 5.2 Add a dense hidden layer with 128 units & ReLu activated.
   - 5.3 Add a dropout layer with rate 0.2
   - 5.4 Add an output dense layer with 10 units and softmax activation.
6. Complete the model using model.compile
7. Train the model on training data.
8. Evaluating the model using model.evaluate.
9. load new set of images and make predictions on them using model.predict
10. STOP

Result: Program executed successfully.

Exp. No: 4 Date 30/10/23 Page No: 5

Experiment - 4

Aim: Implement a python program for classification of a image in CIFAR-10 dataset using neural network

Algorithm

1. Load the CIFAR-10 dataset which contains 60000, 32x32 colour images in 10 different classes and split it.
2. Normalize pixel values to between 0-81, convert label to one-hot encoding
3. Define a neural net model with ~~Adam strategies~~ a Flatten Layer & 3 dense layers with specified hidden unit & Activation fundtin
4. Compile the model with Adam optimizer, categorical cross entropy loss and accuracy metric
5. Train the model on test data and for 5 epochs.
6. Evaluate the model on test data and store the test accuracy.
7. store relevant information about the model in a dictionary.

Exp. No: Date / / Page No: 6

8. Display information about each run & run with highest test accuracy.

9. Make predictions on a few sample images & plot a probability meter.

Result: Program executed Successfully.

30/11/22

Exp. No: 5 Date 30/10/23 Page No: 7

Experiment - 5

Weight Initialisation

Aim:- To compute the impact of Kaiming and Xavier weight initialization techniques on a neural networks performance during training and validation using CIFAR-10 dataset.

Algorithm

1. Import necessary libraries and load CIFAR-10 dataset
2. Normalize the pixel values to range [0,1]
3. Build a neural network model
4. Initialize weights using Xavier & Kaiming initializer
5. Compile models using Adam Optimizer, categorical cross entropy loss & accuracy as a metric
6. Train both models & store training history
7. Evaluate the performance
8. Plot and display accuracy for both models

Result: Program executed successfully

Exp. No: 6 Date 6/10/23 Page No: 8

Experiment - 6

Optimiser Analysis

Aim: To Analyze the impact of optimization technique and visualize the change in performance.

Algorithm

1. Load the MNIST dataset, preprocess and normalize the pixel values
2. Define dictionaries for optimizer and accuracy.
3. For each optimizer
   - 3.1 Build neural network model
   - 3.2 Compile the model
   - 3.3 Train the model on training data.
   - 3.4 Evaluate the model on test data.
4. Plot the training and testing accuracy for each optimizer.

Result

Program executed successfully.

Exp. No: 7 Date 13/10/23 Page No: 9

Experiment - 7

Digit classification using VGG-net 19

Aim: Digit classification using pre-trained networks like VGG-net-19 for MNIST dataset. Analyse and visualize performance improvement.

Algorithm

1. Load and Preprocess MNIST Data
   1.1 Load MNIST Dataset
   1.2 Reshape and normalize.
   1.3 Convert Grayscale to RGB
2. Convert Categorical labels to one hot encoded format.
3. Load Pre-trained VGG-19 model.
4. Build Modified VGG-19 like model
   4.1 Create new sequential model
   4.2 Add pretrained VGG model as base.
   4.3 Flatten & add Dense, Dropout layer.
5. Compile the Model.
6. Train the model.
7. Evaluate the model by making predictions on random test images
8. Display the visualization.

Result: Program executed successfully.

Exp. No: 8 Date 27/10/23 Page No: 10

Experiment - 8

RNN for Review classification

Aim: Implement a simple RNN for review classification on IMDB dataset.

Algorithm

1. Define RNN model
   1.1 create RNN model using Sequential model.
   1.2 Add Embedding, SimpleRNN, Dense layers.
   1.3 Compile the model
2. Load IMDB data
3. Training & Evaluation Function
   3.1 Train the model for specific epochs & batch size.
   3.2 20% validation split to be used.
   3.3 Evaluate the model on test data
   3.4 return loss & Accuracy.
4. Visualize the change in loss, Accuracy along with predictions.

Result: Program executed successfully. 12/12/23

Exp. No: 9 Date 20/11/23 Page No: 11

Experiment - 9

LSTM & GRU for Review Classification

Aim: Implement LSTM and GRU for review classification on IMDb dataset.

Algorithm

1. Import necessary libraries
   1.1 NumPy, Tensorflow, IMDb dataset, Matplotlib
2. Load IMDb Dataset
3. Define Model Building functions
   3.1 Functions for building LSTM, GRU
   3.2 Each model consists of embedding layer, Recurrent layer, and dense output layer
   3.3 Compile the models using binary cross-entropy loss and accuracy metric
4. Define Training and evaluation function
5. Define plotting function to plot training history
6. In Main script define parameters such as max_feature and max length.
7. Print test accuracies of LSTM and GRU models.
8. For each model type, display two examples, showing the review, actual label, predicted probability, and predicted label.

Result: Program executed successfully

Exp. No: 10 Date 27/11/23 Page No:

Experiment - 10

Aim: ANN based Image Classifier

Algorithm

1. Import libraries
2. data = load_data()
3. Split into training and testing data.
4. Normalise image value to range (0,1)
5. Create sequential network model using keras:
   - 5.1 Add a flatten layer to convert 28x28 images to 1D
   - 5.2 Add a dense hidden layer with 128 units.
   - 5.3 Add a dropout layer with rate of 0.2 to prevent overfitting
   - 5.4 Add a output dense layer with 10 units and softmax activation.
6. Compile the model with Adam optimizer
7. Train the model
8. Evaluate and Find the accuracy of the model.

Result: Program executed and output obtained successfully.

SYAMA