(B. Tech.) Semester-VII AY 2023-24

DL Lab Assignment No. 06

Student Name: Rohit Saini_____ **PRN No.: 1032200897_____**

Date: 25-11-23 **Faculty: Prof. Anita Gunjal**

Problem Statement: To study and implement the simple Neural Network for predicting output (Use dataset of Assignment 1).

Objectives:

1. To understand the architecture of ANN.
2. To study & implement ANN.

Theory:

Simple Neural Network (ANN):

A Simple Neural Network, or Artificial Neural Network (ANN), is a computational model inspired by the human brain. It consists of interconnected nodes, or neurons, organized into layers—input, hidden, and output layers. Neural networks learn patterns and relationships in data by adjusting weights and biases during training.

Activation Functions:

Activation functions introduce non-linearities to neural networks, allowing them to model complex relationships in data. Common activation functions include the sigmoid, hyperbolic tangent (tanh), rectified linear unit (ReLU), and softmax. These functions determine the output of each neuron and contribute to the network's ability to learn and generalize.

Loss Function:

The loss function, or cost function, measures the disparity between predicted and actual values. It serves as a guide for the neural network during training, helping it adjust its parameters to minimize this difference. Different tasks require different loss functions. For instance, mean squared error is often used for regression, while binary cross entropy and categorical cross entropy are common for classification.

Gradient Descent Algorithm:

Gradient Descent is an optimization algorithm employed to minimize the loss function by adjusting the parameters of the model. The algorithm iteratively updates weights and biases in the direction that reduces the loss. This iterative process involves computing the gradient of the loss with respect to the parameters and updating them accordingly. Variants of gradient descent, such as Stochastic Gradient Descent (SGD) and Mini-batch Gradient Descent, enhance the efficiency and convergence speed of the optimization process.

Operations to be performed:

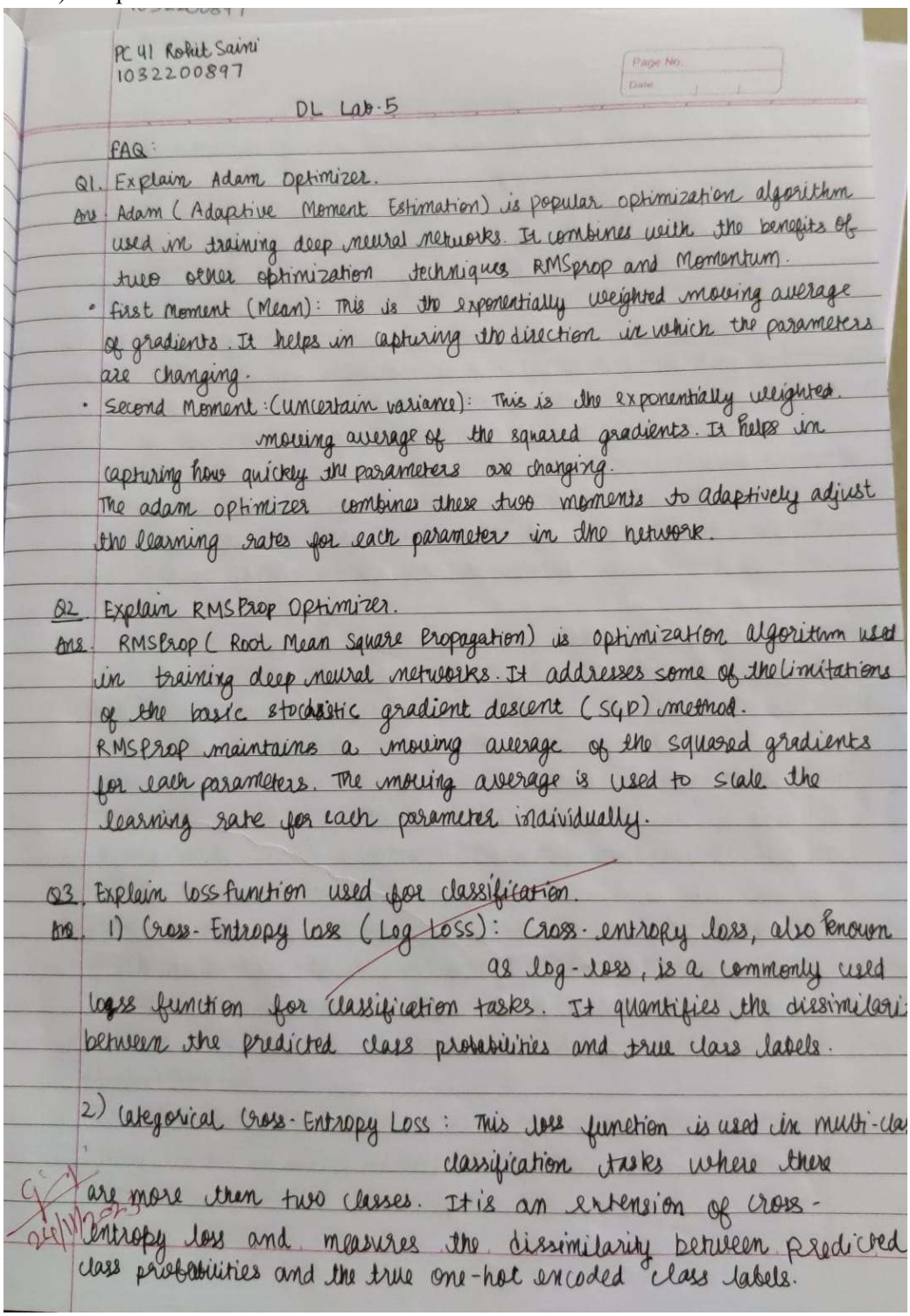
1) Import the required Python libraries
2) Initialize neural network parameters (weights, bias) and define model hyperparameters (number of iterations, learning rate)
3) Train the learning model
4) Plot Loss value vs Epoch
5) Test the model performance

Program code: **(paste your program code)**

Output: **(paste output screen & graphs plotted)**

FAQs:

1) Explain Adam Optimizer.
2) Explain RMSProp Optimizer.
3) Explain loss Function used for classification.



PC 41 Rohit Saini
1032200897

DL Lab-5

FAQ:

Q1. Explain Adam Optimizer.

Ans. Adam (Adaptive Moment Estimation) is popular optimization algorithm used in training deep neural networks. It combines with the benefits of two other optimization techniques RMSprop and Momentum.

- First Moment (Mean): This is the exponentially weighted moving average of gradients. It helps in capturing the direction in which the parameters are changing.
- Second Moment: (Uncertain variance): This is the exponentially weighted moving average of the squared gradients. It helps in capturing how quickly the parameters are changing.

The adam optimizer combines these two moments to adaptively adjust the learning rates for each parameter in the network.

Q2. Explain RMSProp Optimizer.

Ans. RMSProp (Root Mean Square Propagation) is optimization algorithm used in training deep neural networks. It addresses some of the limitations of the basic stochastic gradient descent (SGD) method.
RMSProp maintains a moving average of the squared gradients for each parameters. The moving average is used to scale the learning rate for each parameter individually.

Q3. Explain loss function used for classification.

Ans. 1) Cross-Entropy loss (Log Loss): Cross-entropy loss, also known as log-loss, is a commonly used loss function for classification tasks. It quantifies the dissimilarity between the predicted class probabilities and true class labels.

2) Categorical Cross-Entropy Loss: This loss function is used in multi-class classification tasks where there are more than two classes. It is an extension of cross-entropy loss and measures the dissimilarity between predicted class probabilities and the true one-hot encoded class labels.

24/11/23

Conclusion:

The ANN was implemented and performed successfully for predicting output.