School of Computer Science Engineering and Technology

Course-B. Tech	**Type-** General Elective
Course Code- CSET-335	**Course Name-** Deep Leaning
Year- 2024	**Semester-** Even
Date- 12/03/2024	**Batch-** 2023-2024

CO-Mapping

Exp. No.	Name	CO1	CO2	CO3
06	**Implementing Momentum, RMSProp, and Adam optimizer etc.**	✓	✓	--

Objectives

CO1: To explain the fundamentals of deep learning, Convolution neural network.
CO2: To articulate different problem of classification, detection, segmentation, generation and understand existing solutions/ deep learning architectures.
CO3: To implement a solution for the given problem and improve it using various methods transfer learning, hyperparameter optimization.

Assignment-6

Goal: Implementing Momentum, RMSProp, Adam optimizer and Demonstrating overfitting and implementation of regularizationtechniques: L1, L2, Dropout.

Please find the link of the dataset below:
https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html
https://www.kaggle.com/datasets/uciml/iris

To Do:

- Download and import the data set named "**Iris**" through sklearn. (Use load_iris())
- Slice the data set into **independent and dependent features.**
- Convert the categorical output label to numeric using **one hot encoding.** (to_categorical() function in keras). Apply **standardization** on the independent features. (StandardScaler() function in sklearn)
- Split the data set into train and test.
- Build the NN model using **Sequential model**. (In keras) considering 3 hidden layers with **tanh** activation function in first 2 hidden layers and **relu** in the last one.Use **tanh** in the input layer and

softmax in the output layer.

- Compile the model using **Adam optimizer** and **categorical_crossentropy** loss function.
- Fit the model for **200 epochs and batch size 512.**
- Evaluate the model and print achieved loss and accuracy.
- Visualize the plot between loss and epoch for training and test data.
- **Implement the above code without regularization, L1 regularization, L2 regularization and Dropout**. To implement L1 and L2 regularization, change the above pointand use kernel_regularizer() function on all input and hidden layers. To implement Dropout, change the above point and use Dropout() function (in keras) to drop 20% of nodes on all hidden layers.

Note: For all regularization techniques, visualize the plot between loss and epoch fortraining and test data.

Scenario: Consider any neural network containing a single weight w_0 and bias b_0. Initialize them to some random values using a function.

To Do:

1. Initialize iteration num t to 1.

2. For the given quadratic loss function f(m) = m^2 – 2m +1, write the code to implement Momentum, RMSPRop and Adam optimizers from scratch to update weights until convergence.

Note: In each optimizer, also do the bias correction after computing accumulated gradient equation.

3. The criteria for convergence should be $w^{t-1} = w^t$ (that is when weight at iteration t-1 equals weight at iteration t)

4. At each iteration print the updated weight value and loss value.

5. Plot the visualization showing epoch vs loss for each optimizer: Momentum, RMSProp, and Adam using Matplotlib or Seaborn library function.

6. Finally compare how many iterations it takes to converge in each optimizer and finally conclude with best optimizer (based on loss value) and number of iterationsrequired in that optimizer.