Deep Learning Programming Assignment

Ex. No.	**Date**	**Program** (Tentative)
1	30/01/24	1. Single Layer Perceptron
2	08/02/24	1. Feed Forward & Back-Propagation Learning Algorithm for Multiple Perceptron
3	20/02/24	1. Linear Regression
4	27/02/24	1. ANN for CIFAR10
5	24/03/24	1. CNN for IRIS Flower
7	02/04/24	1. Classifying Skin Lesions using ResNet-101
8	09/04/24	1.
9	16/04/24	1.
10	23/04/24	1.
11	30/04/24	1.

Ex No. 1 Single Layer Perceptron
Date: 29-01-2024

NOTE - Do not use inbuilt functions for perceptron.

1. Implement the Perceptron algorithm from scratch in Python.
    - Initialize the weights with [0 0 0] and a learning rate of 0.0001.
    - For each iteration, calculate the output of the Perceptron for each input in the training set.
    - Use MSE to computer the error for all samples
    - Update the weights using the gradient descent procedure.
    - Repeat the above steps until the Perceptron converges or a maximum number of iterations is reached.
    - Test the trained Perceptron on a separate test set.
    - Use the step function as an activation function in the output layer

Use the IRIS Dataset for the above, considering all four features: sepal length, sepal width, petal length, and petal width, but only two classes - **Setosa, and Versicolor**. Drop the feature vectors of the other class.

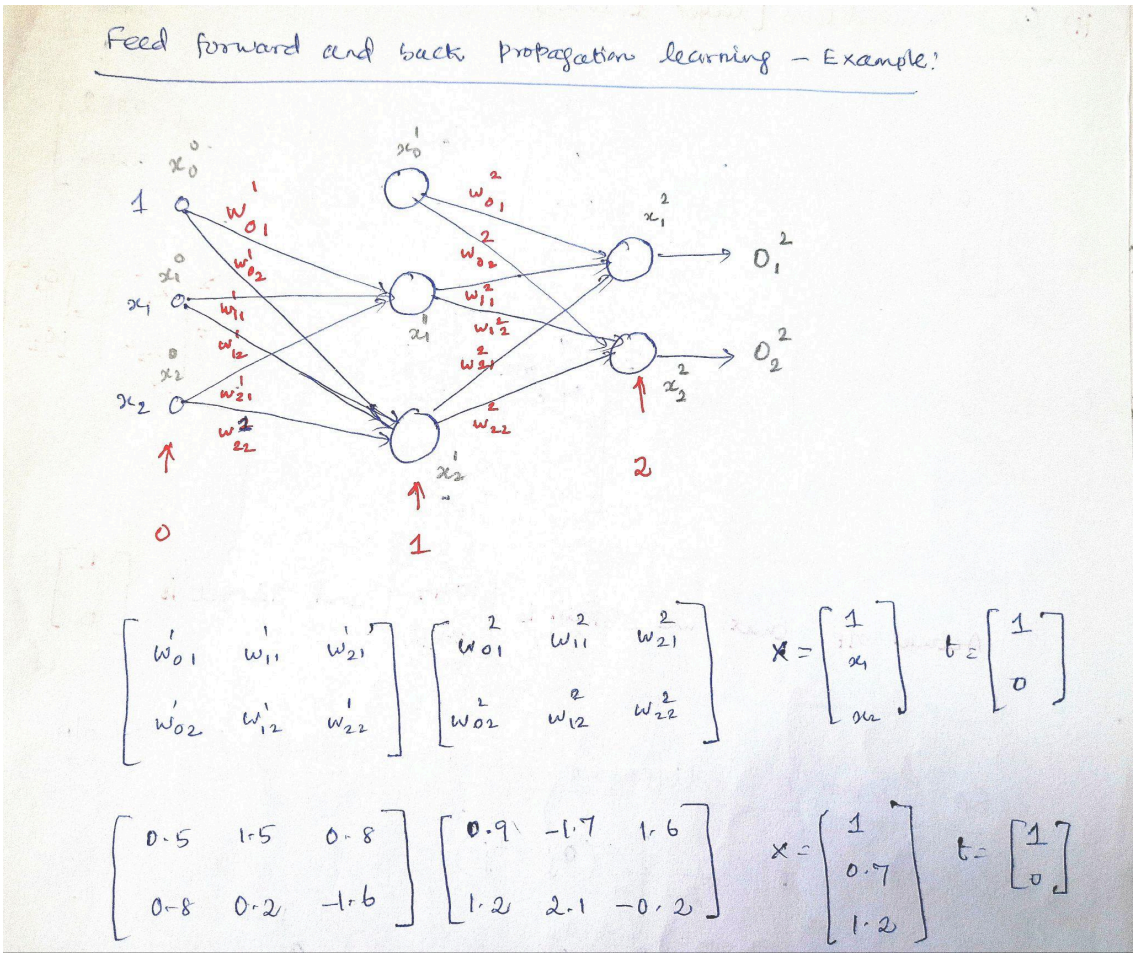
Please find the dataset here - Iris Dataset

Ex No. 2: Feed Forward & Back-Propagation Learning Algorithm for Multiple Perceptron

Date: 08-02-2024

NOTE - Do not use inbuilt functions.

1. Implement the feedforward and backpropagation learning algorithm for multiple perceptrons in Python for the question provided in the attached image.
    a. Initialize the weights and biases randomly.
    b. Implement the forward pass.
    c. Compute the loss between the predicted output and the actual output using an appropriate loss function.
    d. Compute the gradients of the loss function with respect to the weights and biases using the chain rule.
    e. Update the weights and biases.
    f. Iterate over multiple times (epochs), performing forward propagation, loss calculation, backpropagation, and parameter updates in each iteration till convergence.



Ex No. 3 Linear Regression

Date: 14-02-2024

NOTE :- Do not use the inbuilt function for implementing Linear Regression.

Dataset - Linear Regression Dataset

1. Implement the linear regression model from scratch using gradient descent.
    - Start with initializing the parameters m (slope) and c (y-intercept) to zero.
    - Define the loss function as Mean Squared Error (MSE).
    - Calculate the gradients of the loss function with respect to m and c.
    - Update the parameters m and c using the gradients and a learning rate.
    - Iterate the above steps for a fixed number of iterations or until convergence.
    - Plot the cost function over iterations and observe if it is decreasing.
    - After the model is trained, predict the output for a given input and compare it with the actual output.

Ex No. 4 Identifying Images From the CIFAR-10 Dataset Using ANNs

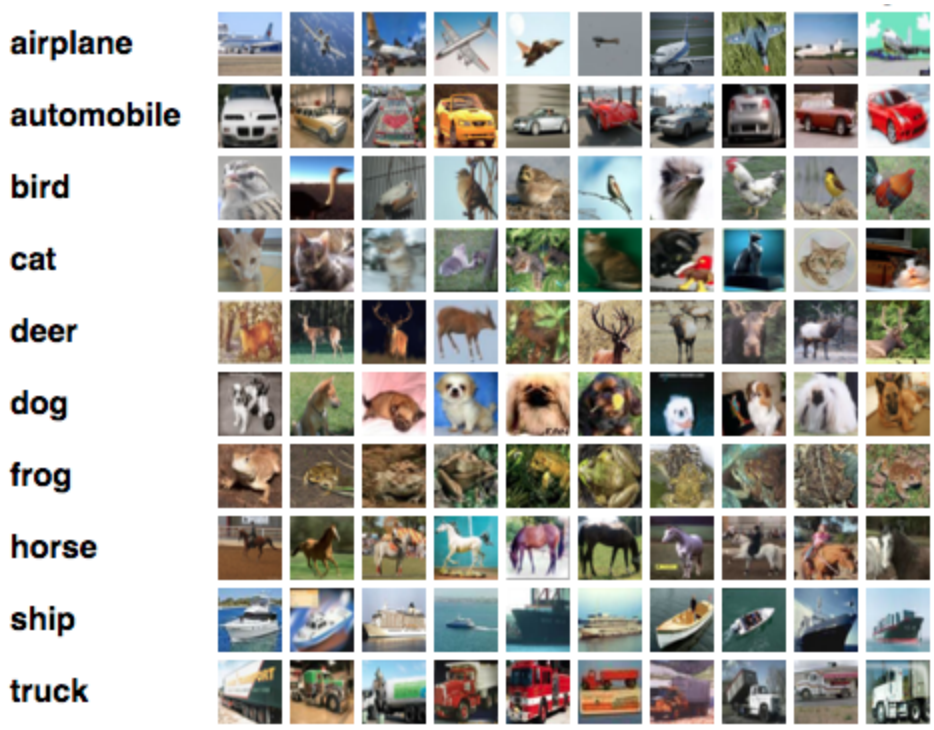
Date: 24-02-2024

NOTE :- Kindly follow the instructions as given in the Tutorial.

Dataset - Can be downloaded from keras or pytorch directly similar to MNIST.

1. Implement the ANN for classifying the images from CIFAR-10 dataset. Steps to be followed are as follows -
    a. Flatten the input image dimensions to 1D (width pixels x height pixels)
    b. Normalize the image pixel values (divide by 255)
    c. One-Hot Encode the categorical column
    d. Build a model architecture (Sequential) with Dense layers(Fully connected layers)
    e. Train the model and make predictions

Information about the dataset -



The CIFAR-10 dataset consists of 60,000 32 x 32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images.

The important points that distinguish this dataset from MNIST are:

- Images are colored in CIFAR-10 as compared to the black-and-white texture of MNIST
- Each image is 32 x 32 pixel
- 50,000 training images and 10,000 testing images

Now, these images are taken in varying lighting conditions and at different angles, and since these are colored images, you will see that there are many variations in the color itself of similar objects (for example, the color of ocean water).

Ex No. 5 Classifying Flowers From the IRIS Flowers Image Dataset Using CNNs

Date: 24-03-2024

NOTE :- Kindly follow the instructions as given in the Tutorial.

Dataset - Present in the classroom assignment.

1. Implement the CNN for classifying the flowers from IRIS Flowers Image dataset.
    a. Implement a custom CNN using Conv2D layers (the configuration of the model can be followed from the tutorial or can be changed as well)
    b. Implement the Transfer Learning Method for VGG16.

    Information Regarding the dataset -

    - The .zip file contains Three folders namely iris-setosa, iris-versicolor and iris-virginica.
    - Each of the images present in that folder belong to that class.
    - Pay attention to the shape of the images present in the dataset and the size which you mentioned while creating the model.

Ex No. 6 Classifying Skin Lesions using ResNet-101

Date: 02-04-2024

1. In this lab assignment, you will be implementing the ResNet-101 model, a deep convolutional neural network, for classifying skin lesions from the ISIC 2018 dataset. Additionally, you will explore the effects of dilation on the performance of the ResNet-101 model for this task.
    a. Download the ISIC-2018 dataset and the Models from the link given at the end of the question.
    b. The images are already preprocessed, prepare the labels properly.
    c. First train the model without dilation and utilize optimization techniques such as regularization, early stopping, etc.
    d. Utilize the proper loss function.
    e. After training the model, evaluate the model's performance on the test dataset.
    f. Calculate classification metrics such as **accuracy, precision, recall, and F1-score.**
    g. Now repeat the same steps for Dilated ResNet-101 and **compare the results for both in the form of a table**.
    h. Further modifications can be done in the model to improve the results such as replacing the GlobalAveragePooling with a series of Dense layers, Changing the dilation rate, etc.

    The student's code with the highest performing metrics will be posted in the Classroom for reference.

ResNet Models - 🗀 ResNet Models
Dataset for Roll Numbers - CS20B1001 to CS20B1113, CS21B1002 to CS21B1054 : isic2018-10k-mv-384-jpg-stratified | Kaggle
Dataset for Roll Numbers - CS21B1056 to CS21B1088, CS21B2001 to CS21B2045 : isic2018-7.5k-non-mv-384-jpg-stratified | Kaggle