Implementation of Multilayer Perceptron (MLP) using NumPy

Reza Mosavi 400222100

16 mars 2024

1 Introduction

General Explanation of a Neural Network Lesson

Neural networks are a class of machine learning models inspired by the structure and function of the human brain. They are powerful tools for solving complex tasks such as classification, regression, and pattern recognition. In a neural network lesson, students typically learn about the fundamental concepts, architecture, and training procedures of these models.

General Structure of the Network

In a neural network lesson, students are introduced to the basic structure of a neural network, which typically consists of three main types of layers:

1. **Input Layer:**
    - The input layer receives the input data, which could be features extracted from images, text, or any other form of structured data.
    - Each neuron in the input layer represents a distinct feature of the input data.
2. **Hidden Layers:**
    - The hidden layers are intermediate layers between the input and output layers.
    - Each hidden layer comprises multiple neurons, where each neuron performs a weighted sum of its inputs followed by the application of an activation function.
    - The activation function commonly used in hidden layers is the Rectified Linear Unit (ReLU), which introduces non-linearity to the network and enables it to learn complex relationships in the data.

3. **Output Layer:**
    - The output layer produces the final predictions or classifications based on the processed information from the hidden layers.
    - The number of neurons in the output layer depends on the nature of the task. For example, for binary classification tasks, a single neuron suffices to output the probability of belonging to one class, while for multi-class classification tasks, the number of output neurons equals the number of distinct classes.

General Procedure of the Code

In a neural network lesson, students often learn about the general procedure involved in building and training a neural network. This includes:

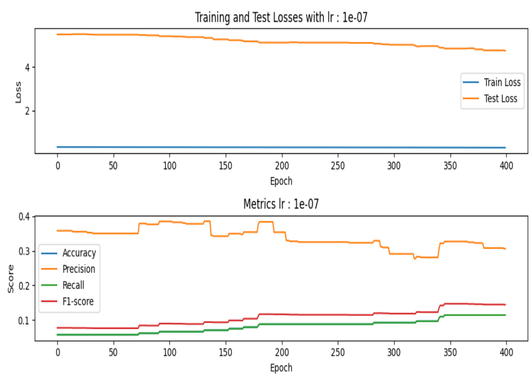
1. **Initialization:**
    - The network parameters, including the number of layers, number of neurons in each layer, and activation functions, are initialized.
    - Weights and biases are initialized randomly or using specific initialization techniques.
2. **Forward Pass:**
    - Input data is passed through the network, propagating forward from the input layer through the hidden layers to the output layer.
    - Matrix operations are utilized for efficient computation, avoiding the need for explicit loops and enhancing computational speed.
    - Neurons in each layer perform a weighted sum of their inputs, followed by the application of the activation function (e.g., ReLU).
3. **Backpropagation:**
    - The network's predictions are compared with the actual targets, and the error is calculated.
    - Using matrix calculus and vectorization, gradients of the loss function with respect to the network parameters (weights and biases) are efficiently computed.
    - The gradients are then used to update the parameters using optimization algorithms such as gradient descent.
4. **Training:**
    - The network is trained using labeled training data.
    - During training, the forward pass and backpropagation procedures are repeated iteratively for multiple epochs, allowing the network to learn from the data and adjust its parameters to minimize the loss.

5. **Evaluation:**

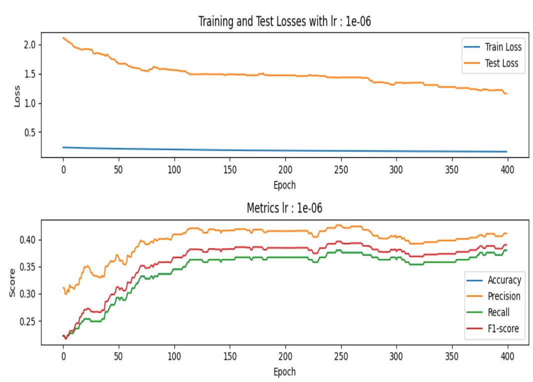
   - Once trained, the network's performance is evaluated on a separate validation or test dataset to assess its generalization ability.
   - Evaluation metrics such as accuracy, precision, recall, and F1-score are computed to quantify the model's performance.

By understanding the general structure and procedure of a neural network, students gain insight into how these models operate and how they can be applied to various real-world problems in machine learning and artificial intelligence.

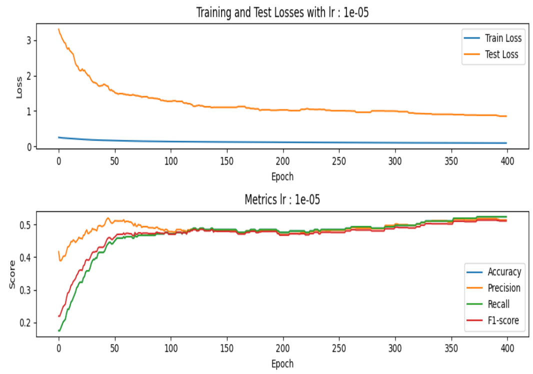
2 Examining different learning rates



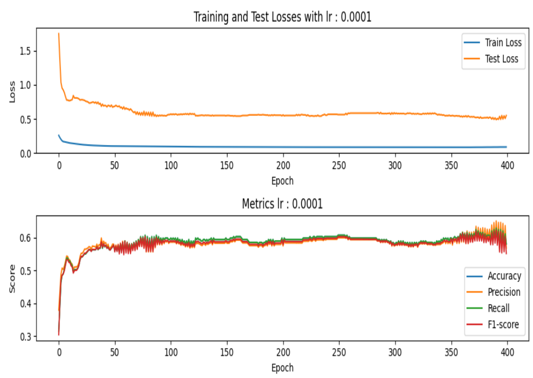
Figur 1: lr:0000001



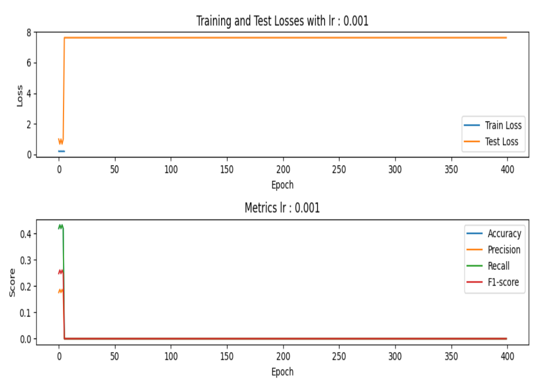
Figur 2: lr:000001



Figur 3: lr:00001



Figur 4: lr:0001

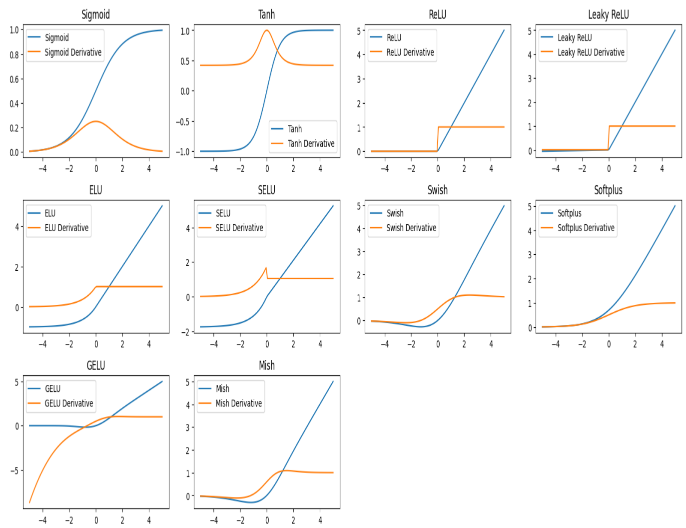


Figur 5: lr:001

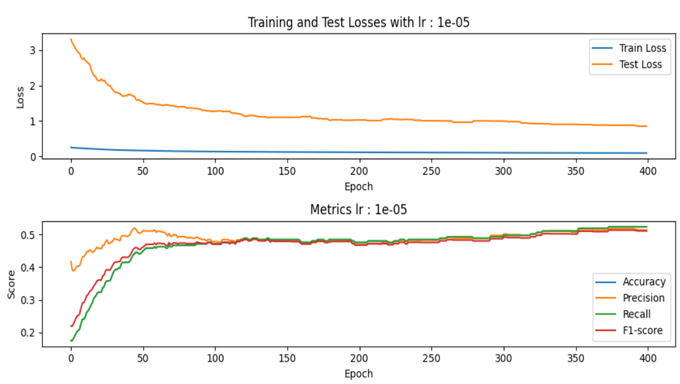
It is clear that the value of the learning rate coefficient is very critical and it only works well in a general range, it loses its effect from a smaller value and more than 0.001 generally disrupts the learning process.

3 Activation Functions

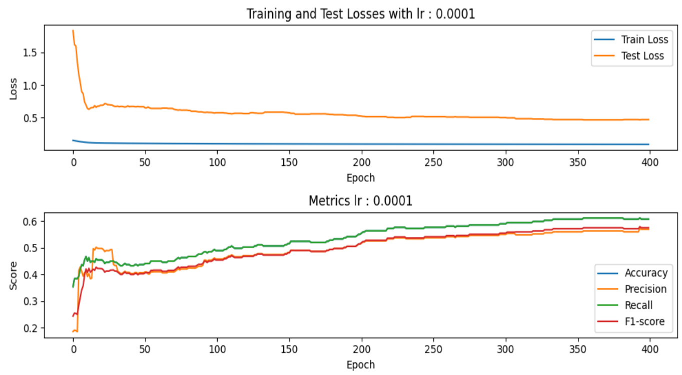
Now, before examining the ten activation functions on the neural network and comparing their states, let's have a general look at their graphs and their derivative graphs.



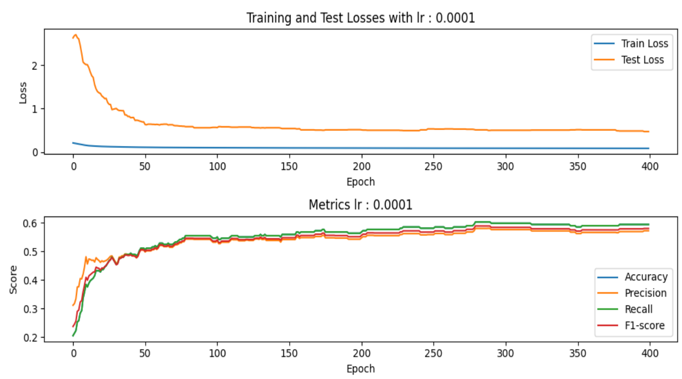
Figur 6: Activation Functions



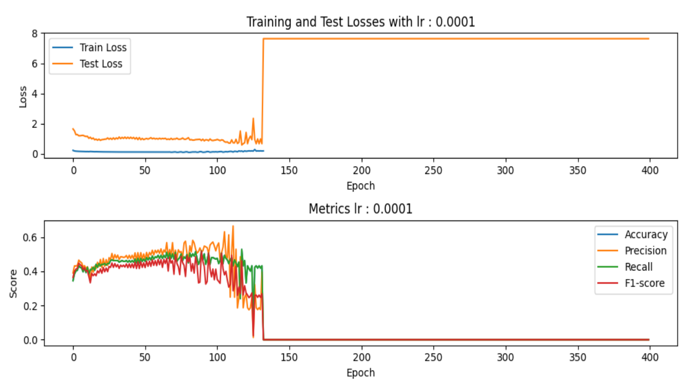
Figur 7: ReLU Activation Functions



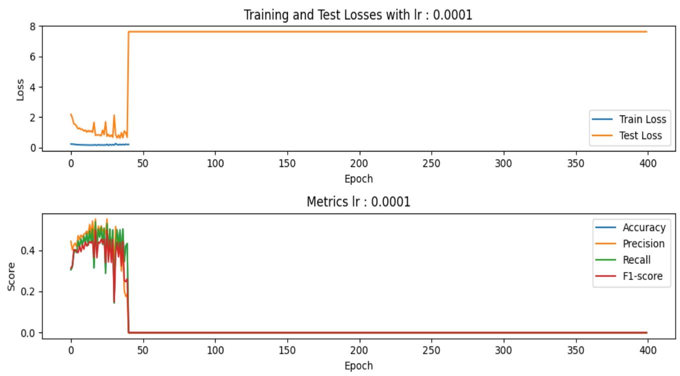
Figur 8: Sigmoid Activation Functions



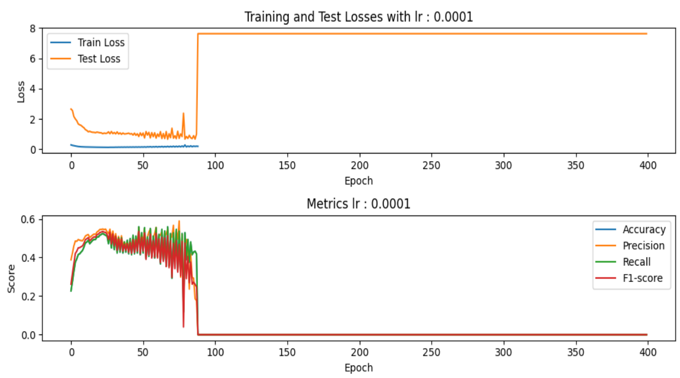
Figur 9: Tanh Activation Functions



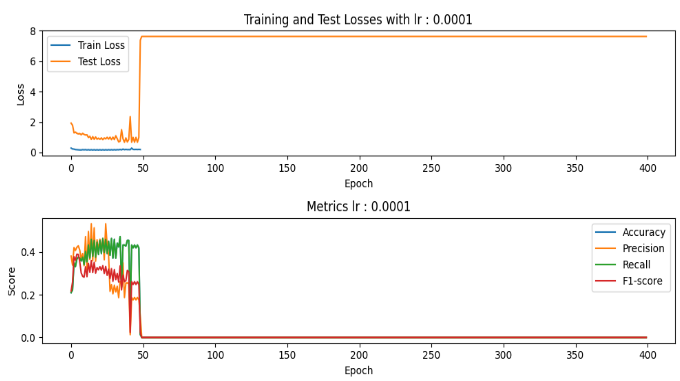
Figur 10: Leaky ReLU Activation Functions



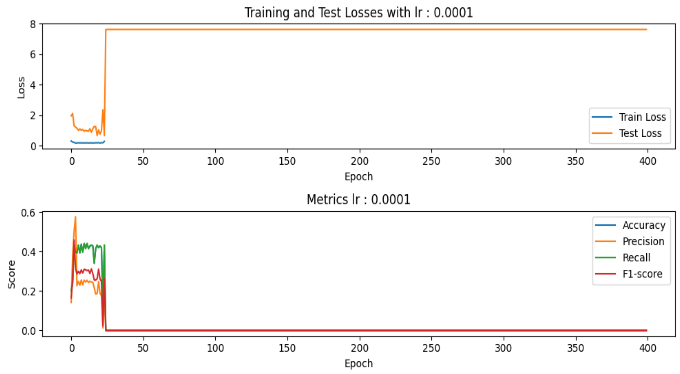
Figur 11: ELU Activation Functions



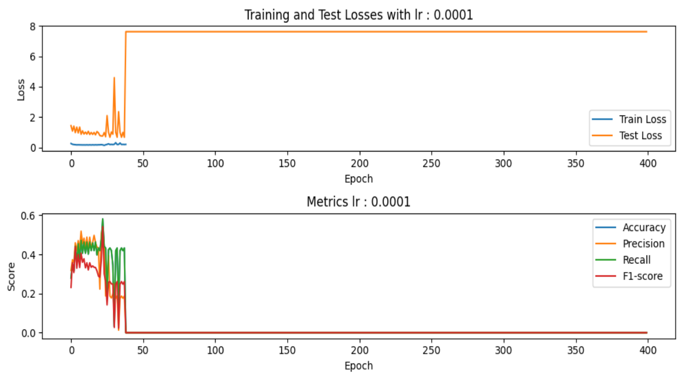
Figur 12: SeLU Activation Functions



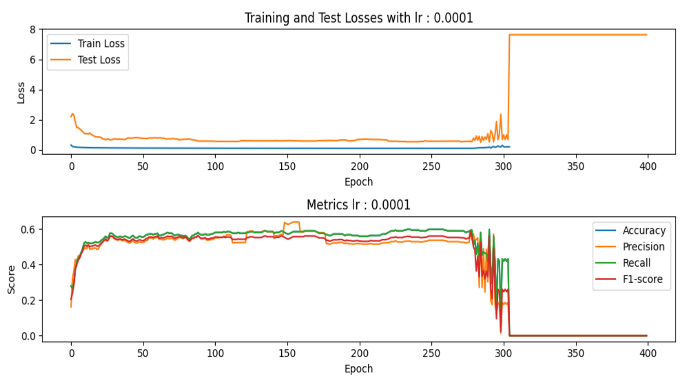
Figur 13: Swish Activation Functions



Figur 14: Softplus Activation Functions



Figur 15: Gelu Activation Functions



Figur 16: Mish Activation Functions

As is clear, in some of the activator functions, due to the structure of the derivative graph, they can converge much faster, but in any case, with all these activator functions, the results can be reached, but with more epochs!