Technical Report: Custom Activation in MLP for Classification Problem

Introduction

In this technical report, I present the implementation details of a Multilayer Perceptron (MLP) model with a custom activation function for the task of classification. The custom activation function is defined as a linear function of the form k0 + k1 * input, where k0 and k1 are learnable parameters. We will discuss the algorithm, initial settings, parameter updates, final parameter values, train and test performance metrics, and visualizations of the loss function over epochs over Iris Dataset. We will also discuss the evaluation parameters.

Algorithm Overview

- Loading the Iris dataset and splitting it into training and test sets.
- Preprocessing the input features by normalizing them using MinMaxScaler.
- Encoding the target labels using one-hot encoding.
- Defining the custom activation function as a subclass of tf.keras.layers.Layer.
- Building the MLP model with the custom activation function.
- Compiling the model with the Adam optimizer and categorical cross-entropy loss.
- Defining a custom callback to print the values of k0 and k1 at the end of each epoch.
- Training the model on the training set for a specified number of epochs.
- Evaluating the model on the test set and printing the test loss and accuracy.
- Retrieving the learned values of k0 and k1 from the model.
- Generating predictions on the test set and converting one-hot encoded labels back to original labels.
- Computing and printing the classification report (including F1-score) based on the predictions.
- Plotting the training and validation accuracy over epochs.
- Plotting the training and validation loss over epochs.

Initial Settings

The Iris dataset is loaded using the load_iris function from sklearn.datasets. The input features (X) and target labels (y) are extracted from the dataset. The shape of X is (150, 4), indicating 150 samples with 4 features each.

The input features are split into training and test sets using the train_test_split function from sklearn.model_selection. The test set size is set to 20% of the total samples, and a random seed of 42 is used for reproducibility.

The input features are normalized using MinMaxScaler from sklearn.preprocessing. The scaler is fitted on the training set and then applied to both the training and test sets.

The target labels are encoded using one-hot encoding with OneHotEncoder from sklearn.preprocessing. The encoding is applied separately to the training and test sets.

A Custom Activation Function is defined as a subclass of tf.keras.layers.Layer. The activation function takes the inputs and applies the linear transformation k0 + k1 * input. The learnable parameters k0 and k1 initialized with he_normal distribution are defined as trainable weights using add_weight in the constructor. The activation function is implemented in the call method.

Initial k0: 0.07098616659641266

Initial k1: [0.42404717, -0.17590643, -0.10915639, 0.3652909]

Building the MLP Model

The MLP model is built using tf.keras.models.Sequential. It consists of two dense layers. The first dense layer has 10 units and uses the custom activation function defined earlier. A dropout layer with a dropout rate of 0.4 is added after the first dense layer. The second dense layer has 3 units and uses the softmax activation function for multiclass classification. The model is compiled using the Adam optimizer with a learning rate of 0.01. The loss function is set to 'categorical_crossentropy', and the accuracy metric is tracked during training. A CustomCallback is defined to print the values of k0 and k1 at the end of each epoch. The callback accesses the model's layers and weights to retrieve the values.

Prediction and Classification Report

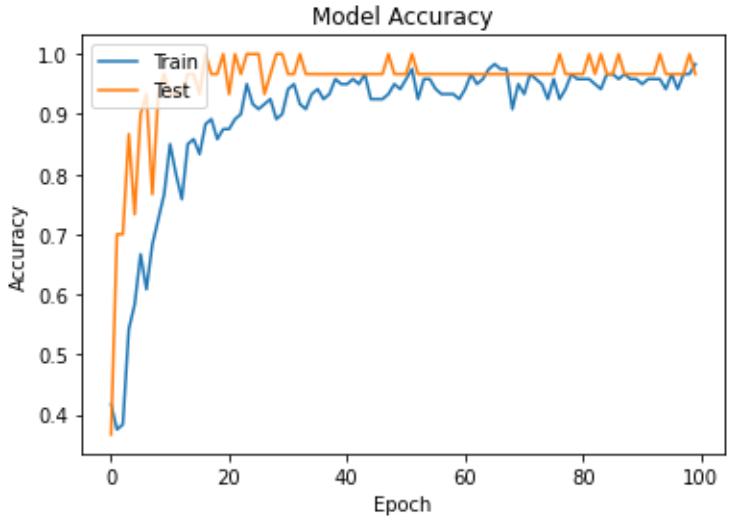
Final k0: 0.52130234

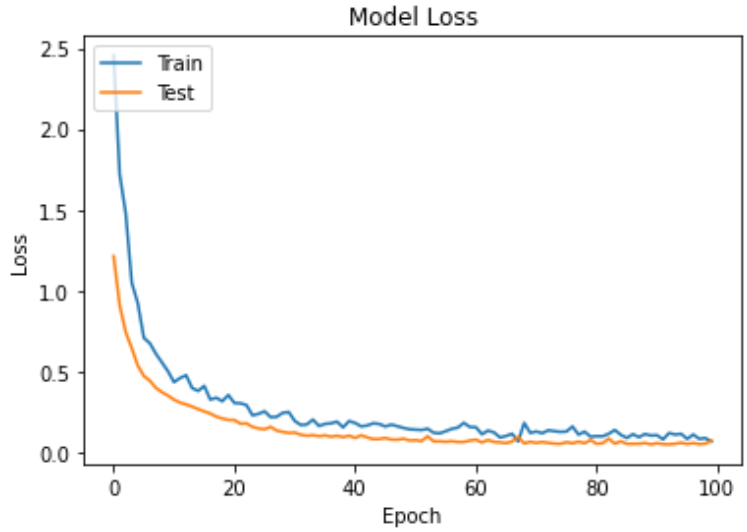
Final k1: [0.15964708, 0.66050684, -1.2817582 , -1.0792134]

Test Loss: 0.07342558354139328

Test Accuracy: 0.9666666388511658

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	0.89	0.94	9
2	0.92	1.00	0.96	11
accuracy			0.97	30
macro avg	0.97	0.96	0.97	30
weighted avg	0.97	0.97	0.97	30





Loss and Accuracy Breast Cancer Dataset

Test Loss: 0.060534846037626266

Test Accuracy: 0.9736841917037964

Loss and Accuracy Bank Note Dataset

Test Loss: 0.04034900292754173

Test Accuracy: 0.9927272796630859

Loss and Accuracy MNIST Dataset

Test Loss: 0.2941182851791382

Test Accuracy: 0.919700026512146