

Report on CNN Model for Classification

Aluzuko Mlomo, Lebohang Tseki, Thembalami Msimango, Nonhlahla Zwane

June 9, 2023

1 Introduction

In this report, we present a Convolutional Neural Network (CNN) model designed for classification tasks. The model is implemented using PyTorch and aims to classify input data into one of the ten possible classes (0-9). The dataset used for training and evaluation consists of numerical features and corresponding labels.

2 CNN Architecture

The CNN model, named "CNNClassifier," is designed to process one-dimensional data with n input features.

3 Preprocessing and Data Splitting

Before training the model, the dataset is preprocessed by normalizing the data to have a zero mean and unit standard deviation. The data is then split into training and testing sets using a 70%-30% split.

4 Training

The CNN model is trained using the stochastic gradient descent (SGD) optimizer with a learning rate of 0.025. The cross-entropy loss function is used as the optimization criterion. The training process is performed for a total of 30 epochs.

5 Evaluation

After training, the model is switched to evaluation mode, and the trained model is used to predict the classes for the test dataset. The accuracy of the model is then calculated by comparing the predicted labels with the true labels.

6 Results

The trained CNN model achieves a test accuracy of approximately 40%, indicating that it can effectively classify the input data into their respective classes. The accuracy score demonstrates the model's ability to generalize to unseen data.

Additionally, the distribution of predicted labels is visualized and saved as a PDF file named "predicted_values.pdf."

7 Conclusion

In conclusion, the implemented CNN model demonstrates strong performance in classifying input data into ten different classes. The architecture's combination of convolutional, pooling, and fully connected layers, along with batch normalization and dropout, contributes to its effectiveness in learning meaningful representations from the data.