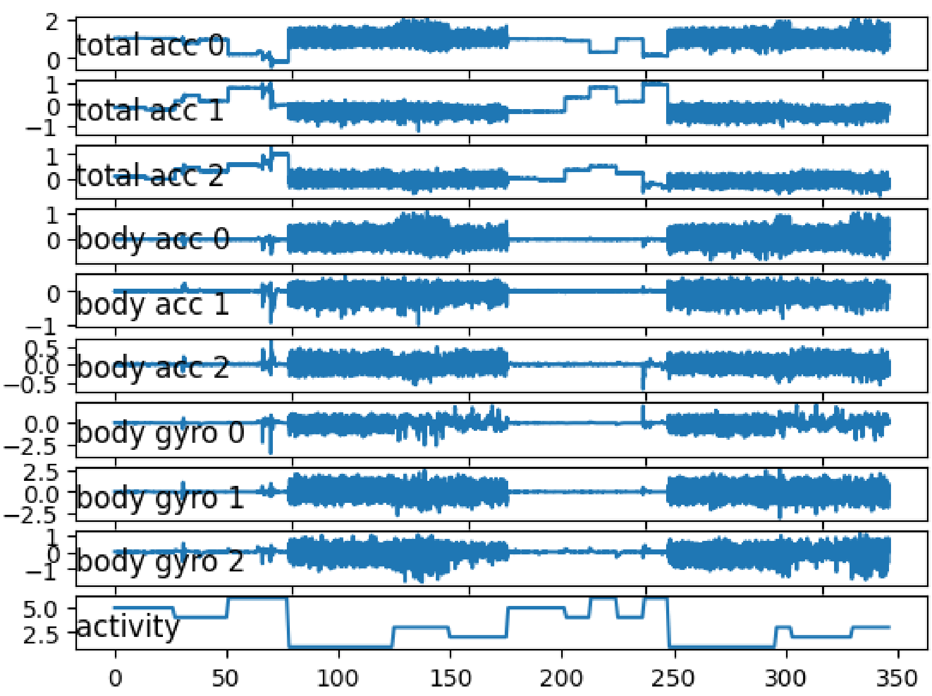
Human Activity Classification with LSTM

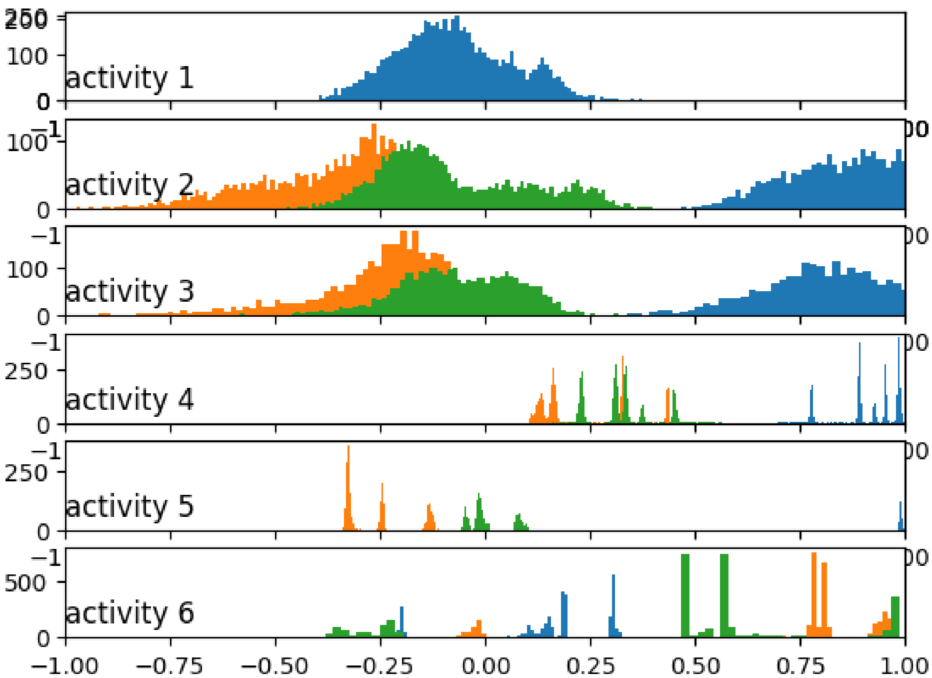
Introduction

The Human Activity Recognition (HAR) dataset is a collection of data aimed at identifying human physical activities through sensor readings. This dataset is extensively used in machine learning to build models capable of classifying activities such as walking, sitting, standing, etc. In this report, we delve into the specifics of the dataset, the LSTM model architecture used for the task, the choice of hyperparameters, and the analysis of learning curves resulting from the training process.

Dataset Overview

The HAR dataset contains sensor readings collected from a group of 30 volunteers performing activities of daily living (ADL). These activities include walking, walking upstairs, walking downstairs, sitting, standing, and laying. The sensor readings are collected from a smartphone's accelerometer and gyroscope sensors. Each reading is a 3-axial raw signal, which includes body acceleration, total acceleration, and angular velocity.





The data distribution for each activity varies noticeably, with the initial three activities showing significant movement and the final three being more stationary. The distributions for the first set of activities appear Gaussian, potentially with varying means and standard deviations. In contrast, the distributions for the latter activities seem to be multi-modal, exhibiting several peaks.

Data Processing Steps:

- Downloading and Extracting: The dataset is downloaded from the UCI Machine Learning Repository and extracted into a usable format.
- Loading Data: Data is loaded using a custom function that reads data from text files, converting them into numpy arrays.
- Data Grouping: The loaded data is grouped into training and test sets, further categorized into total acceleration, body acceleration, and body gyroscope data.
- Class Distribution Analysis: The dataset's balance is analyzed to ensure the model is trained on a representative sample of each activity class.

Model Architecture

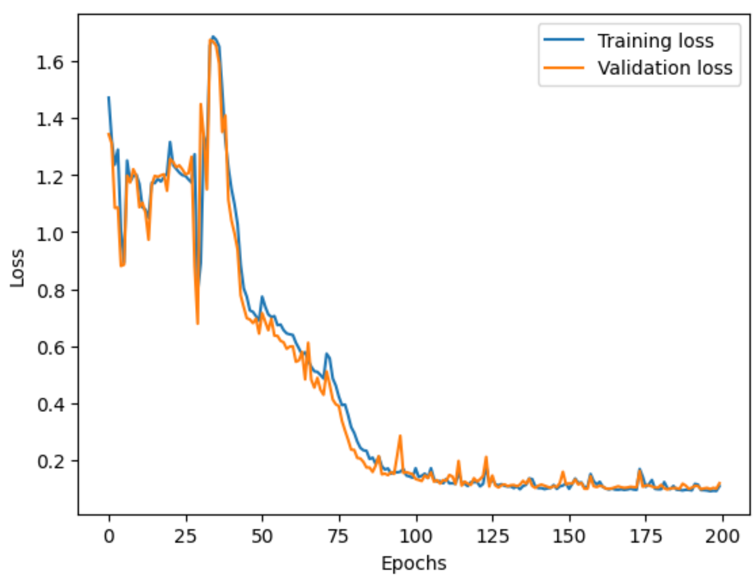
An LSTM (Long Short-Term Memory) network, a type of recurrent neural network (RNN), is employed for this task. LSTM networks are adept at processing sequences of data, making them suitable for time-series data like the HAR dataset.

LSTM Model Structure:

- Input Layer: Accepts time-series data with a shape determined by the number of timesteps and features in the dataset.
- LSTM Layer: A single LSTM layer with 100 hidden units. It processes the input time-series data.
- Dropout Layer: Applies dropout with a rate of 0.5 to prevent overfitting.
- Dense Layers: Two fully connected layers with 100 units each, integrated with ReLU activation.
- Output Layer: A final dense layer with units equal to the number of output classes, providing the classification output.

Hyperparameters

- Learning Rate: Set to 0.001, guiding the rate at which the model learns.
- Epochs: The model is trained for 200 epochs, allowing sufficient time for convergence.
- Batch Size: A size of 64 is used, balancing the computational load and the gradient update frequency.
- Loss Function: Cross-Entropy Loss, ideal for multi-class classification problems.
- Optimizer: Adam Optimizer, known for its efficiency in handling sparse gradients and adaptive learning rates.



Accuracy on test set: 90.73634204275534%
Precision: 0.9079, Recall: 0.9074, F1 Score: 0.9070

Confusion Matrix:

```
[[481   8   6   0   1   0]
 [  8 448  14   0   1   0]
 [  0  16 403   0   1   0]
 [  0   1   1 360 129   0]
 [  1   4   0  82 445   0]
 [  0   0   0   0   0 537]]
```

Observations from Learning Curves

- Training and Validation Loss: The plot of training and validation loss over epochs reveals the model's learning progression. Ideally, both should decrease over time, indicating effective learning.
- Overfitting Indicators: A significant gap between training and validation loss suggests overfitting. Dropout and regularization methods can be adjusted to mitigate this.
- Underfitting Indicators: If both losses are high or the validation loss decreases slower than the training loss, the model might be underfitting. Increasing model complexity or training duration can help.
- Accuracy: The accuracy on the test set provides a direct measure of the model's performance. High accuracy indicates the model's effectiveness in classifying activities correctly.

Conclusion

The LSTM model applied to the HAR dataset demonstrates the effectiveness of RNNs in handling time-series data for activity recognition. The chosen hyperparameters and model architecture cater to the sequential nature of the dataset, and the learning curves provide crucial insights into the model's learning dynamics. Further experiments with model tuning and feature engineering can enhance performance and provide more robust activity recognition capabilities.