SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

Faculty of Engineering

Re-implementation and Optimization of a Fall Prediction Model Based on the IoMT Framework

Rathnayaka M.H.M - EN21474132

Abstract

Falls are a leading cause of injuries among the elderly, making their prediction and prevention a critical area of research. This research focuses on re-implementing and optimizing the fall prediction model presented in the paper *cStick: A Calm Stick for Fall Prediction, Detection, and Control in the IoMT Framework*. The model, designed to predict and detect falls using physiological and environmental data, was originally trained on a dataset of 9,670 samples. However, only a subset of 2,040 samples was publicly available for this implementation.

The original model was recreated with a fully connected neural network using six input features, two hidden layers with 20 units each, and sigmoid activation for classification into three target classes. Challenges occurred due to overfitting, driven by the reduced dataset size and the model's complexity. To address this, the model was optimized by introducing L2 regularization, dropout layers, a simplified architecture (8 and 4 units in the hidden layers), and the Adam optimizer with a reduced learning rate.

The optimized model achieved improved generalization and validation accuracy, demonstrating its adaptability to smaller datasets. These findings highlight the importance of tailoring deep learning models to real-world constraints and their potential integration into IoMT devices like the cStick for elderly healthcare.

1. Introduction

Falls are a significant concern for the elderly, often leading to severe injuries, loss of independence, and reduced quality of life. According to global statistics, over 37 million falls occur annually among people aged 65 and older, with many resulting in hospitalization or death. Fall-related injuries impose a substantial burden on healthcare systems and families. Preventive measures, including early prediction and timely intervention, are essential to mitigate these risks. With advancements in technology, Internet of Medical Things (IoMT) devices have emerged as promising solutions to enhance elderly care through real-time monitoring and intelligent predictions.

The research paper *cStick: A Calm Stick for Fall Prediction, Detection, and Control in the IoMT Framework* proposes an innovative IoMT-enabled device that integrates multiple physiological and environmental parameters to predict and detect falls. The cStick system leverages sensors to monitor heart rate variability, accelerometer data, blood sugar levels, SpO2, grasping pressure, and proximity to nearby objects. The original study employed a fully connected neural network trained on a dataset of 9,670 samples, achieving a test accuracy of approximately 96.67%.

This research aims to re-implement the fall prediction model described in the paper while addressing the challenges posed by the publicly available subset of the dataset, consisting of only 2,040 samples. The objective is to replicate the model, optimize it for smaller datasets, and evaluate its performance.

2. Methodology

2.1 Dataset

The dataset used in this study is a subset of the original dataset presented in the paper *cStick: A Calm Stick for Fall Prediction, Detection, and Control in the IoMT Framework*. While the original dataset contained 9,670 samples, the publicly available subset comprised only 2,040 samples. Each data point included six key features that capture physiological and environmental signals:

- Heart Rate Variability (HRV): Sudden changes in heart rate as an indicator of instability.
- Accelerometer Readings: Linear acceleration data indicating body movement.
- Blood Sugar Levels: Low levels associated with falls due to weakness or unconsciousness.
- Blood Oxygen Levels (SpO2): Low oxygen saturation is linked to dizziness and fainting.
- Grasping Pressure: Variations in grip strength as a fall sign.
- Distance to Nearest Object: Proximity to obstacles or supports during movement.

The target variable had three classes:

1) No Fall: Normal activity without signs of instability.
2) Fall Prediction: Warning signs of an imminent fall.
3) Fall Detection: A confirmed fall event.

2.2 Original Model

The original model employed a fully connected neural network designed for multi-class classification. The architecture included six features representing physiological and environmental data in the input layer. Hidden Layers: Two dense layers with 20 units each, using the ReLU activation function for non-linearity. Output Layer: Three units with sigmoid activation, providing probabilities for the three target classes.

Training parameters were set as follows:

- Optimizer: Stochastic Gradient Descent (SGD) with a learning rate of 0.01.
- Loss Function: Categorical cross-entropy, suitable for multi-class problems.
- Epochs: 200, And Batch Size: 32 samples per batch.

Despite achieving high accuracy in the original implementation, the model faced challenges when applied to a smaller dataset.

```
# Define the model
model = Sequential([
    Dense(20, input_dim=6, activation='relu'),
    Dense(20, activation='relu'),
    Dense(3, activation='sigmoid')
])

# Compile the model
model.compile(loss='categorical_crossentropy',
              optimizer=SGD(learning_rate=0.01),
              metrics=['accuracy'])
```

Figure 1 - original model architecture

2.3 Challenges with the Smaller Dataset

The reduced dataset size introduced significant constraints, primarily:

- Overfitting: The model memorized the training data instead of generalizing it to unseen data.
- High Variance: Predictions fluctuated across epochs, reflecting instability in learning.
- Insufficient Samples: The dataset lacked sufficient diversity to capture the full range of fall scenarios, making it challenging for the model to learn robust patterns.

These challenges highlighted the need for optimization to make the model suitable for smaller datasets.

2.4 Optimizations

To address the challenges, several modifications were made to the original model:

1. Regularization: Added L2 regularization with a factor of 0.05 to penalize large weights and reduce overfitting.
2. Dropout: Introduced a 20% dropout rate to the hidden layers, randomly deactivating neurons during training. This forced the model to learn redundant representations and improved generalization.
3. Simplified Architecture: Reduced the hidden layers to 8 units and 4 units, respectively, to align the model's complexity with the dataset size.
4. Optimizer Adjustment: Replaced SGD with the Adam optimizer at a learning rate of 0.0001. Adam dynamically adjusts learning rates during training, improving convergence and stability.
5. Early Stopping: Implemented a custom early stopping mechanism to terminate training at epoch 34 when validation performance stopped improving, preventing overtraining.

These optimizations significantly improved the model's ability to generalize from the smaller dataset, enabling more stable and accurate predictions.

```
# Model with regularization
model = Sequential([
    Dense(8, input_dim=6, activation='relu', kernel_regularizer=l2(0.05)),
    Dropout(0.2),
    Dense(4, activation='relu', kernel_regularizer=l2(0.05)),
    Dropout(0.2),
    Dense(3, activation='softmax')
])

# Compile the model
model.compile(
    optimizer=Adam(learning_rate=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

Figure 2 - Modified model architecture

3. Results

3.1 Performance Comparison

The re-implemented original model and the optimized model were evaluated based on their performance on the dataset. The key metrics compared include test accuracy, generalization capability, and stability during training.

Metric	**Original Model**	**Optimized Model**
Hidden Layers	2 (20 units each)	2 (8 and 4 units)
Optimizer	SGD (0.01 learning rate)	Adam (0.0001 learning rate)
Regularization	None	L2 Regularization (factor 0.05)
Dropout	None	20% dropout per layer
Early Stopping	None	Stopped at epoch 34
Test Accuracy (%)	96.67% (in original), 100% (overfit)	97.06%
Validation Loss	High (overfitting)	Low (better generalization)

3.2 Accuracy and Loss Analysis

- Original Model: The original model exhibited signs of overfitting, achieving an unrealistic 100% training accuracy, which failed to generalize to unseen data. The training loss approached zero, while the validation loss plateaued, indicating a lack of robustness. These results highlighted the necessity of using regularization techniques and simplifying the architecture to address the overfitting issue effectively.

- Optimized Model: The optimized model demonstrated better generalization capabilities. Validation accuracy is closely aligned with training accuracy, and overfitting was significantly reduced through the application of dropout and L2 regularization. Early stopping played a critical role in preventing overtraining, as training was halted at the optimal epoch based on validation performance.

- The optimized model's higher validation accuracy (97.06%) and lower validation loss underscore the improvements in stability and generalization. These enhancements validated the efficacy of the modifications, particularly the use of the Adam optimizer, regularization, and dropout.

This comparative analysis confirms that the optimizations successfully addressed the limitations of the original model while achieving improved performance metrics.

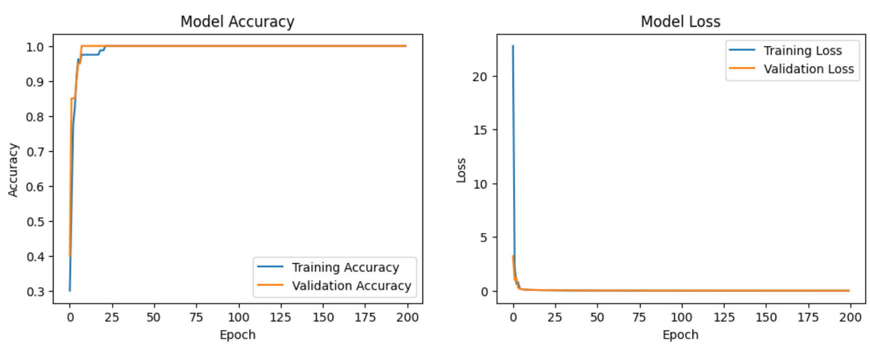

Figure 3 - original model training/ validation loss and accuracy plot

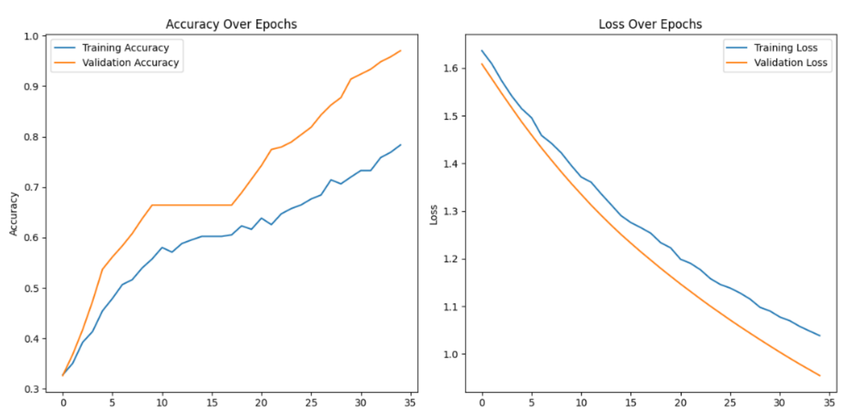

Figure 4 - modified model training/ validation loss and accuracy plot

4. Discussion

4.1 Insights from Re-implementation

Re-implementing the original model highlighted its tendency to overfit due to the lack of regularization, dropout, and adaptive learning rates. These limitations impacted its ability to generalize to unseen data.

4.2 Impact of Optimizations

The optimizations, simplified architecture, dropout, L2 regularization, Adam optimizer, and early stopping improved the model's generalization, reducing overfitting. The optimized model demonstrated better validation accuracy and lower validation loss, ensuring more stable performance.

4.3 Potential Applications in IoMT Devices

The optimized model's enhanced generalization makes it suitable for IoMT devices like the cStick, where reliable fall prediction is critical for user safety.

5. conclusion

This project aimed to re-implement and optimize a fall prediction model, addressing its limitations to improve performance and generalization. The re-implementation of the original model revealed significant overfitting due to the absence of regularization, dropout, and adaptive optimizers. In contrast, the optimized model successfully mitigated these issues through the inclusion of regularization techniques, dropout layers, the Adam optimizer, and early stopping.

The key findings demonstrate that the optimized model achieved a higher validation accuracy (97.06%) and reduced validation loss compared to the original implementation. These improvements underscore the value of thoughtful architectural and hyperparameter adjustments in enhancing model robustness and applicability.

The contributions of this work extend beyond academic interest, offering practical value for real-world applications, particularly in IoMT devices like the cStick. By ensuring reliable and accurate predictions, the optimized model can enhance the safety and usability of wearable medical devices.

In conclusion, this study not only highlights the critical role of optimizations in addressing model limitations but also sets the foundation for future advancements in fall prediction systems.