Universidade NOVA de Lisboa

Information Management School

Predicting Activty Type Using 1D Convolutional Neural Networks

Deep Learning Neural Networks

Author:
Celso Cassamá
Student ID: 20222138

Author:
Guilherme Silva
Student ID: 20221209

Author:
Murilo Oliveira
Student ID: 20222137

Author:
José Matias
Student ID: m2014043

2024

1 Introduction

The project's goal is the creation of a neural network designed to predict types of activities using physiological data gathered from Inertial Movement Units (IMUs). Activities such as lying down, sitting, standing, walking, vacuum cleaning, and ironing are encompassed. The data for this study is sourced from the Physical Activity Monitoring for Aging People 2 (PAMAP2) dataset [1], provided by the University of California, Irvine Machine Learning repository.

The primary objective was the development of a model capable of identifying activities by leveraging neural networks, mainly their ability for pattern recognition in data. Secondary objectives include enhancing our knowledge in data processing fields, particularly signal processing and manipulation, and in the utilization of neural networks, specially the proper selection of model architecture and tuning of training parameters.

The report will start by giving a brief overview of the dataset and the data collection process , including the experimental procedure and the problems that arose from this. It will then go over how the model was built, covering the steps like data pre-processing, model construction, training and evaluation procedure. Afterwards, the results will be presented and discussed. Finally, a summary of the report will be given, focusing on the findings and limitations encountered. Moreover, possible future projects derived from this will be presented and briefly discussed.

2 PAMAP2 Dataset

The PAMAP2 Dataset, comprising data from various activities captured via sensors, was developed to aid in the research of physical activity recognition, with a special emphasis on the elderly.

The data was collected from a group of 9 participants (8 males and 1 female), with an average age of 27.2 ± 3.31 years and a BMI of $25.11 \pm 2.62 kg/m^2$. The study protocol required all subjects to complete 12 standard activities and offered 6 additional optional activities.

Data was acquired using 3 IMUs with a sampling frequency of 100Hz, placed on the chest, wrist of the dominant arm and the ankle of the dominant side, as well as a heart rate monitor with a sampling frequency of 9Hz. The IMUs contain 3 sensors each, an accelerometer, a magnetometer and a gyroscope. Each sensor collects data in all 3 axes. Euler angle data was also collected, however it should not be considered per the dataset guidelines.

However, the dataset contains gaps in data, primarily due to two factors: occasional data dropouts from wireless sensors and more prevalent issues with the hardware setup, such as connection losses and software crashes, resulting in missing activity data for some participants.

Due to these data losses, it became important to choose activities that were not only of comparable length, but also performed by the majority of subjects. Figure 1 shows the total seconds of available data by subject and activity, as well as the number of subjects that performed each activity. Activities 1, 2, 3, 4 , 16 and 17 were chosen, as they have comparable amounts of available data and were performed by 8 of the 9 subjects. Moreover each of the 8 subjects performed all of the activities chosen.

The dataset comprises 9 *.dat* files, with each file documenting the activity data of an individual subject. Within these files, there are 54 columns per row, encompassing information such as timestamps, activity ID, heart rate, and IMU data. For the purposes of this project, focus was placed on only 33 columns, corresponding to timestamps, activity ID, heart rate, temperature, and the axial data of each IMU sensor.

Highlighting the limitations of the dataset is crucial, particularly the small number of subjects, which could potentially impede the models' ability to generalize effectively or train adequately. Furthermore, this limited subject count complicates the division of data into training, validation, and testing sets. Achieving a balance that enables both robust model generalization and reliable testing becomes nearly unfeasible with such a small sample size. Furthermore, the dataset presents a challenge in terms of activity length variation; while similar in duration, the disparity between the longest and shortest activities is notable. Such variation demands padding when employing Recurrent Neural Networks (RNNs). This issue, combined with the inherently large size of each time series, might lead

Activities performed by subjects (in seconds)

	subject101	subject102	subject103	subject104	subject105	subject106	subject107	subject108	subject109	Sum	Nr. of subjects
1 – lying	271.86	234.29	220.43	230.46	236.98	233.39	256.1	241.64	0	1925.15	8
2 – sitting	234.79	223.44	287.6	254.91	268.63	230.4	122.81	229.22	0	1851.8	8
3 – standing	217.16	255.75	205.32	247.05	221.31	243.55	257.5	251.59	0	1899.23	8
4 – walking	222.52	325.32	290.35	319.31	320.32	257.2	337.19	315.32	0	2387.53	8
5 – running	212.64	92.37	0	0	246.45	228.24	36.91	165.31	0	981.92	6
6 – cycling	235.74	251.07	0	226.98	245.76	204.85	226.79	254.74	0	1645.93	7
7 – Nordic walking	202.64	297.38	0	275.32	262.7	266.85	287.24	288.87	0	1881	7
9 – watching TV	836.45	0	0	0	0	0	0	0	0	836.45	1
10 – computer work	0	0	0	0	1108.82	617.76	0	687.24	685.49	3099.31	4
11 – car driving	545.18	0	0	0	0	0	0	0	0	545.18	1
12 – ascending stairs	158.88	173.4	103.87	166.92	142.79	132.89	176.44	116.81	0	1172	8
13 – descending stairs	148.97	152.11	152.72	142.83	127.25	112.7	116.16	96.53	0	1049.27	8
16 – vacuum cleaning	229.4	206.82	203.24	200.36	244.44	210.77	215.51	242.91	0	1753.45	8
17 – ironing	235.72	288.79	279.74	249.94	330.33	377.43	294.98	329.89	0	2386.82	8
18 – folding laundry	271.13	0	0	0	0	217.85	0	236.49	273.27	998.74	4
19 – house cleaning	540.88	0	0	0	284.87	287.13	0	416.9	342.05	1871.83	5
20 – playing soccer	0	0	0	0	0	0	0	181.24	287.88	469.12	2
24 – rope jumping	129.11	132.61	0	0	77.32	2.55	0	88.05	63.9	493.54	6
Labeled total	4693.07	2633.35	1743.27	2314.08	4117.97	3623.56	2327.63	4142.75	1652.59	27248.27	
Total	6957.67	4469.99	2528.32	3295.75	5295.54	4917.78	3135.98	5884.41	2019.47	38504.91	

Figure 1: Activities performed by subjects in seconds and number of subjects that performed each activity.

to memory constraints in the model training process.

3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs), originally designed for analyzing 2D data such as images, have been adapted to handle one-dimensional sequential data, marking a significant advancement in time-series analysis. The 1D CNNs operate by applying convolutional operations along the temporal dimension of data, efficiently detecting local temporal patterns within sequences [2]. This adaptation is particularly relevant for datasets where understanding temporal relationships in sensor data is crucial for activity recognition.

The core advantage of 1D CNNs in handling time-series data lies in their ability to capture temporal dependencies and local features along the time axis. This approach is more effective than traditional fully connected networks, which treat input data as flat vectors, lacking the capacity to preserve temporal context [3]. As a result, 1D CNNs offer a nuanced analysis of time-dependent data, an essential feature for accurately classifying activities based on sensor readings.

Furthermore, 1D CNNs are known for their computational efficiency, especially beneficial when working with smaller datasets [4]. Their architecture, characterized by shared weights and fewer parameters, reduces the risk of overfitting and enables the network to learn time-invariant features. This efficiency in learning and generalization makes 1D CNNs a robust tool for real-world applications in activity recognition, leveraging their capability to understand complex temporal data structures.

This approach can be leveraged to simplify activity prediction within the PAMAP2 dataset. It avoids the use of more computationally expensive neural network architectures which, when combined with the large volume of data in each time series, could lead to excessive training times.

4 Model Development

4.1 Data Importing and Formatting

The model development phase begins with the importation and pre-processing of data. This process begins by importing *.dat* files and subsequently generating a dataframe from this data. In adherence to the dataset guidelines, it is necessary to exclude both the orientation data and the accelerometer data that is calibrated to *6g*. Moreover, data with an activity of 0 must also be excluded, as it corresponds to the 'transient activity' between the activities described in the protocol.

As previously mentioned, data loss occurred in some sensor readings due to the inherent challenges associated with wireless IMUs. This missing data was addressed through interpolation. However, even after interpolation, some heart rate values remained absent. These were manually assigned the first non-null value from their respective columns.

4.2 Task Definition

After converting the data into a usable format, an initial analysis was conducted to assist with task definition and focus. The initial step in defining the task involved selecting the activities to be used. The criteria for this selection included having a sufficient amount of data, being performed by the majority of subjects, and possessing similar data volumes compared to other chosen activities.

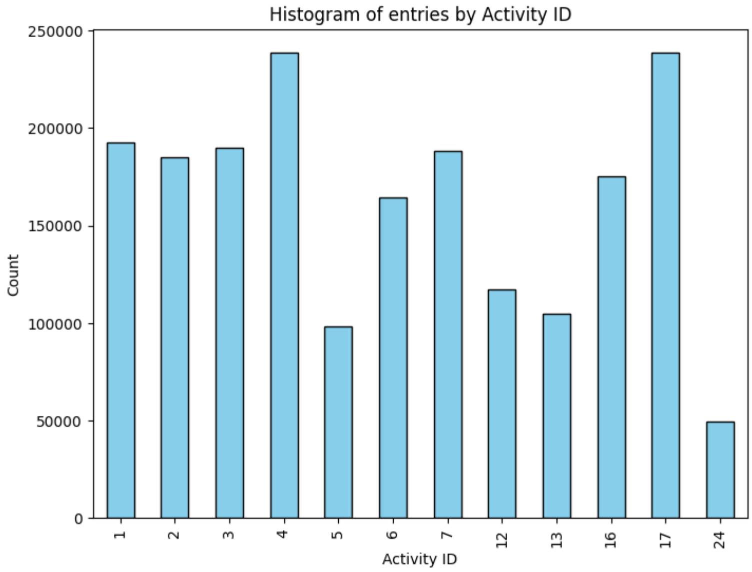

Figure 2: Histogram displaying the distribution of entries by activity ID.

As seen in figure 2, the activities that exhibit similar counts are *1, 2, 3, 4, 6, 7, 16 and 17*. However, and as seen in figure 1, activities *6 and 7* were only performed by 7 subjects. As such, and in accordance to the previously established criteria, the activities chosen for the task were **1, 2, 3, 4, 16 and 17**. These correspond to **lying, sitting, standing, walking, vacuum cleaning and ironing**, which provides a diverse set of activities, likely enhancing the model's predictive capabilities.

With the activity set and the modeling approach established, the task can be defined as: **Predicting activity type from a dataset comprising IMU sensor readings from six varied activities, on a row-by-row basis, using 1D convolutional neural networks.**

4.3 Model Evaluation

Having established the model architecture and the project's objective, the next step is to devise a process for evaluating the model. The goal is to develop a model capable of accurately predicting activity type from the available sensor data on a row-by-row basis. Therefore, it is crucial to employ metrics that effectively assess model quality, especially in scenarios involving classes of varying sizes. The evaluation involved analyzing the model's performance using a comprehensive set of metrics, including accuracy, precision, recall, F1 score, and Cohen's Kappa score. Additionally, confusion matrices were constructed to provide further insights and facilitate comparisons. The metrics were obtained by testing against a previously decided set of subject data.

The chosen metrics for evaluating the model are selected to provide a comprehensive analysis of its performance across various aspects. Accuracy offers a general overview of the model's effectiveness in correctly predicting activity types. However, accuracy alone can be misleading, especially in imbalanced datasets such as the one used. Precision and recall address this while the F1 Score provides a balanced measure of both of them. Cohen's Kappa score is included as it accounts for the possibility of random chance in the model's predictions, offering a more robust assessment of its performance. Confusion matrices were included as they offer a visual representation of the model's performance across different classes, highlighting specific areas of strength or weakness. Lastly, the Area Under the Precision-Recall Curve (AUPRC) is crucial for evaluating performance in datasets with a class imbalance, as it focuses on the model's ability to distinguish between classes under such conditions.

4.4 Model Architecture

The initial versions of the model were basic 1D CNNs with a minimal number of layers. These preliminary models were instrumental in exploring the feasibility of using this neural network type for the given task. Subsequent iterations involved experimenting with different layer combinations to refine the model. The final version includes additional layers aimed at enhancing the model's ability to generalize, as the data obtained from the IMUs have a high degree of variability. It employs a convolutional neural network (CNN) architecture, using 1D convolutions for feature extraction from time-series sensor data. The model's parameters were meticulously optimized through a grid-search method. It is comprised of the following layers:

- **Input Layer**: The input layer receives the data, which is transformed into a 3D tensor with dimensions (*batch_size, sequence_length, sensor_channels*).
- **Conv1D Layers**: Two 1D convolutional layers with filters=128, the number of filters determines the complexity of the features that can be extracted. In this case, using 128 filters allows the model to learn a large number of different features from the sensor data. And *kernel_size=7* are employed, The kernel size determines the size of the window that is used to scan over the input data. These layers apply filters to the input data to extract relevant features, using a kernel size of 7 allows the model to capture both local and global patterns in the sensor data. The activation function ReLu is used to introduce non-linearity. The ReLU function takes a real number as input and outputs either the input itself (if the input is positive) or zero (if the input is negative). This means that the ReLU function introduces a threshold to the model, allowing it to learn non-linear relationships between the input and output.
- **MaxPooling1D Layers**: Two MaxPooling1D layers with *pool_size=2* follow the convolutional layers. These layers reduce the dimensionality of the feature maps by downsampling, allowing for better generalization. Pooling layers downsample the feature maps by removing redundant information, which helps to reduce overfitting and improve computational efficiency. A larger pooling size will remove more information, while a smaller pooling size will preserve more detail. In this case, using a pooling size of 2 provides a good balance between removing redundant information and preserving detail. The MaxPooling1D layers effectively compress the information from the convolutional feature maps into a smaller and more compact representation. This compression is crucial for improving the model's efficiency and generalizability. By focusing on the most important features and discarding redundant information, the model can learn more effectively and generalize better to unseen data.
- **GlobalMaxPooling1D Layer**: A GlobalMaxPooling1D layer aggregates the features from the pooled feature maps, effectively compressing the information into a vector representation. The GlobalMaxPooling1D layer serves several important purposes such as reducing the dimensionality of the feature representation from a spatial representation to a scalar representation. This is beneficial because it can help to reduce overfitting and improve computational efficiency. It also aggregates the information from multiple feature maps into a single vector, which allows the model to learn more global patterns in the data. Finally, it is relatively insensitive to small variations in the input data, which helps to improve the model's generalization ability.

- **Dense Layers**: Two fully connected dense layers with neurons=128 are used to further process the aggregated features extracted by the convolutional layers. The activation function relu is used again. After the MaxPooling1D layers have downsampled the feature maps, the model enters the dense layer stage. These fully connected layers provide a higher level of abstraction, allowing the model to learn more complex relationships between features. In the CNN model for HAR, two dense layers are employed, each with 128 neurons. The ReLU activation function is used again to introduce non-linearity in the model. The dense layers process the aggregated features extracted by the convolutional layers, transforming and combining them to generate a more comprehensive representation of the sensor data.

- **Dropout Layer**: A Dropout layer with rate=0.5 is inserted to prevent overfitting. It randomly drops out half of the connections during training, forcing the network to learn more robust features.

- **Output Layer**: The final output layer consists of 6 neurons, each representing a possible activity category. The activation function softmax is used to ensure the output probabilities sum to one. Each neuron in the output layer computes a score that represents the probability that the input data belongs to the corresponding activity class. The output layer uses the softmax activation function to normalize these scores so that they sum to one. This ensures that the model assigns a probability of 1 to the activity that it is most confident about and assigns probabilities of 0 to the other activity classes.

5 Results and Discussion

As mentioned in section 4.3, the models were evaluated using a set of metrics alongside the respective confusion matrices. The models used for testing were the Early Iteration Model, the Non-Optimized Parameters Model and the Optimized Parameters Model. All models were trained using the stratified k-folds method using *k=5*.

The Early Iteration Model has a similar architecture to the Optimized Parameters Model, but with fewer layers: one less Conv1D and one less MaxPooling1D. This model served to provide insight into the viability of the approach used, as such, it was only ran for 5 epochs.

The Non-Optimized Parameters Model has the same architecture as described in section 4.4 and was trained for a maximum of 50 epochs with a batch size of 32, utilizing the Adam optimizer and categorical crossentropy loss. Early stopping, with a patience set to 3, was implemented based on the validation loss and configured following regular configurations: *filters = 128; size = 7; pool_size = 2; and dense_neurons = 128.*

Finally, the Optimized Parameters Model was constructed by conductive a grid search across various combinations of hyperparameter values. The models were trained for a maximum of 50 epochs with a batch size of 32, utilizing the Adam optimizer and categorical crossentropy loss. Early stopping, with a patience set to 3, was implemented based on the validation loss. The following parameter ranges were considered: *Filters (64, 128); Filter Size (3, 5, 7); Pool Sizes (2, 3); Dense Neurons (64, 128).*

The data in table 1 and the confusion matrix in figure 3 refer to the performance of one of the earlier iterations of the model. It was comprised of fewer layers than the model presented in section 4.4, notably only having a single 1D Convolution layer.

Metric	**Value**
Accuracy	0.7108
Precision	0.7275
Recall	0.7108
F1 Score	0.7037
Cohen's KappaScore	0.6521

Table 1: Evaluation metrics for the Early Iteration Model

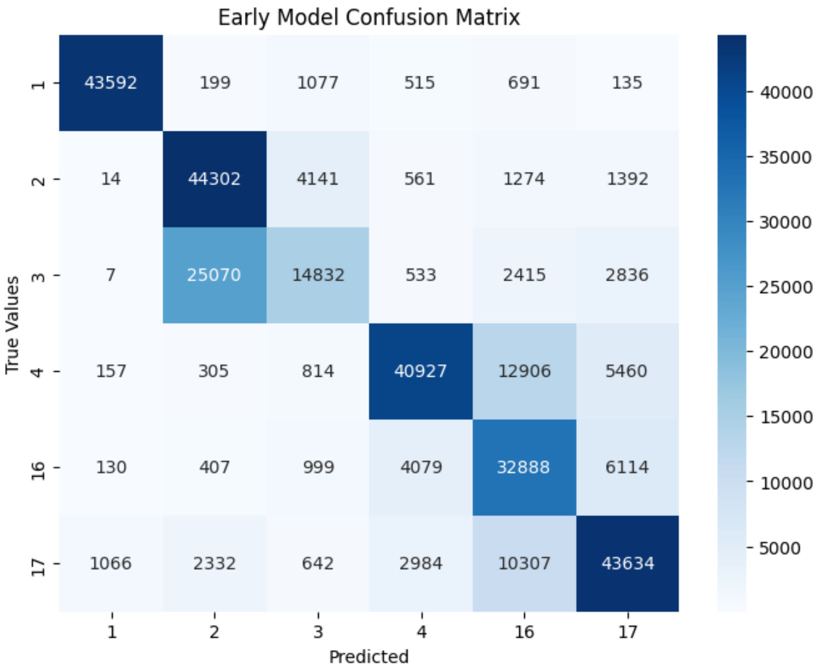

Figure 3: Confusion Matrix for the Early Iteration Model.

The data presented in Table 2 and the confusion matrix shown in Figure 4 depict the performance of a model similar to the final version using non-optimized parameters. While its architecture mirrors the one detailed in Section 4.4, the parameters within each layer have not undergone optimization.

Metric	Value
Accuracy	0.7301
Precision	0.7849
Recall	0.7301
F1 Score	0.7274
Cohen's KappaScore	0.6751

Table 2: Confusion Matrix for the Non-optimized Model.

The results from the model with optimized parameters are presented in table 3, and its corresponding confusion matrix is displayed in figure 5. The architecture of this model is detailed in Section 1, where it is noted that the parameters were optimized using a grid search method.

Metric	Value
Accuracy	0.8430
Precision	0.8521
Recall	0.8430
F1 Score	0.8397
Cohen's KappaScore	0.8109

Table 3: Evaluation metrics for the Final Model with optimized parameters.

The Early Iteration Model exhibited promising results with an accuracy and an F1 score of 0.7037. Notably, its precision and recall were balanced, also at 0.7275 and 0.7108, suggesting that the approach may be viable. The Cohen's Kappa score of 0.7987 indicated a reasonable degree of agreement, taking into account the possibility of the agreement occurring by chance. The confusion matrix for this model

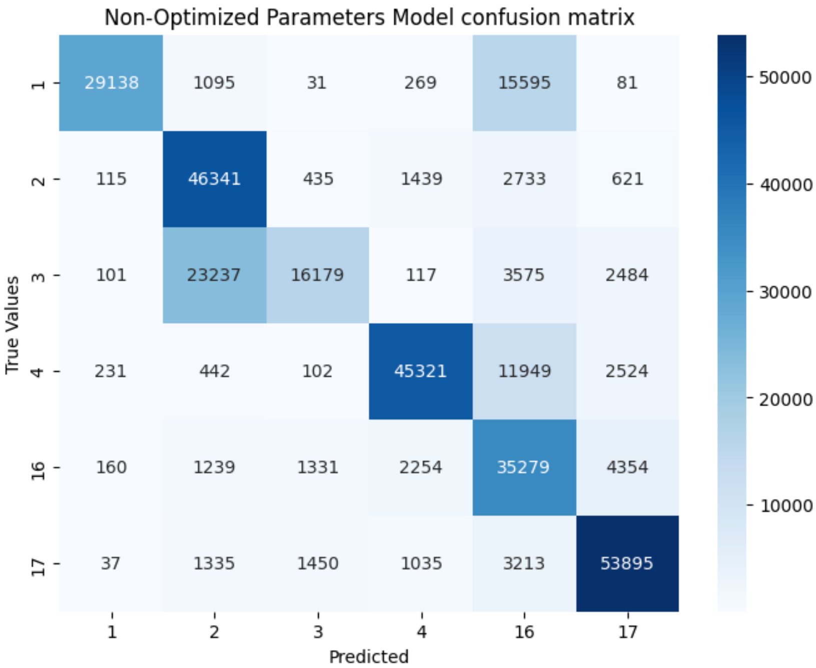

Figure 4: Confusion Matrix for the Non-Optimized Model.

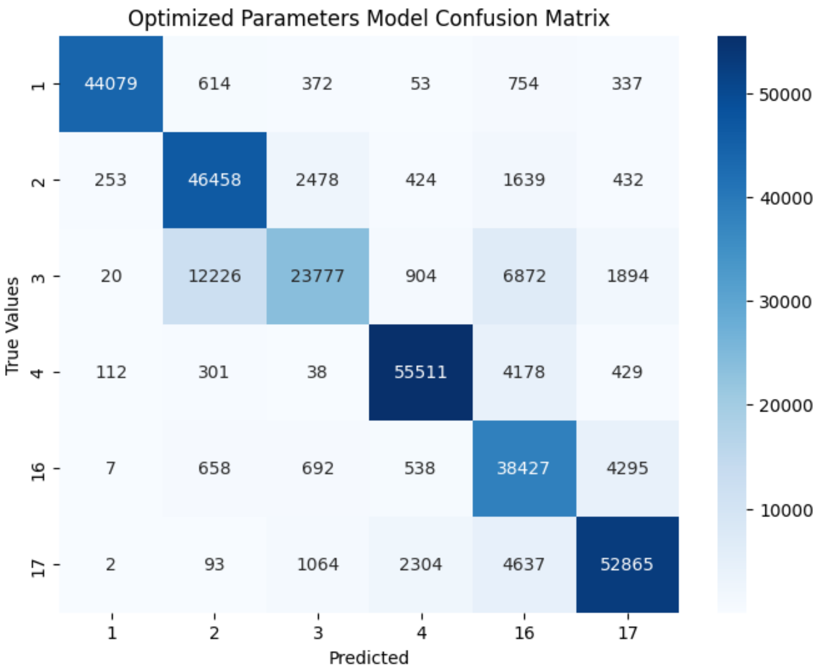

Figure 5: Confusion Matrix for the Optimized Model.

revealed high true positive rates across most classes, with some confusion noticeable between classes 2 and 3. These correspond to sitting and standing, which are activities that elicit similar physiological responses. The results of this early iteration provided some insight into the feasibility of using 1D CNN to predict activity type.

The Non-Optimized Model showed some improvement in all evaluation metrics. The accuracy rose to 0.7301, and the Cohen's Kappa score to 0.6751, pointing to moderate agreement. This suggests that, while the model improved, it may have been overfitting or not generalizing well to the data. The confusion matrix further supported this, with a similar number of misclassifications, particularly noticeable between classes 2 and 3, and classes 4 and 16. As with the previous model, the misclassifications between standing and sitting are attributable to the similar physiological responses. This can also explain the misclassifications between class 4, walking, and 16, vacuum cleaning. These activities involve a greater amount of movement when compared to the remaining classes, which may explain why the model is able to distinguish them from the rest while still misclassifying some cases between them.

Subsequent optimization of the model parameters resulted in the Optimized Model, which demonstrated an improvement over the Non-Optimized Model, with an accuracy of 0.8430 and an F1 score of 0.8397. The precision and recall showed congruence, and the Cohen's Kappa score increased to 0.8109, indicating strong agreement and an effective model calibration. The confusion matrix of the Final Model showed a marked improvement in the correct classification rates, especially for classes that were previously misclassified. This model commits mistakes at a lower rate than the Non-Optimized one. Despite this, the number of misclassifications between classes 2 and 3 still remains high, which indicates that the model is still unable to properly distinguish between the two.

In conclusion, the performance of the Optimized Models shows that the approach chosen may be viable for classifying activity types in real-world scenarios. Moreover, the addition of 1D Convolution and Max Pooling layers proved to lead to better results. The further optimization of the training parameters lead to a substantial improvement in the final model. Overall, the results align with previous expectations, as the increased model complexity and refinement lead to markedly better results. Further training with a larger, more diversified dataset may translate into a model that is even more capable of predicting activty type, even in cases where the movement and physiological responses associated with said activities may be similar.

6 Conclusion

This project successfully explored the adaptation of Convolutional Neural Networks (CNNs) for time-series analysis, specifically focusing on activity recognition using IMU sensor data. The computational efficiency of 1D CNNs, attributed to shared weights and fewer parameters, makes them particularly suitable for real-world applications, especially when dealing with smaller datasets.

In early experimentation, various neural network types were tested, including Recurrent Neural Networks (RNNs) that processed each time series consisting of data from a single activity and subject. However, it was the 1D CNNs that demonstrated greater potential and promise for the task at hand.

In the model development phase, the focus was on the appropriate choice of activities to predict and the most suited model architectures for this intent. The chosen activities for prediction (lying, sitting, standing, walking, vacuum cleaning, and ironing) were selected to provide a set as representative as possible. The model architecture was built using 1D convolutional layers, max pooling and softmax activation function on the output layer after evaluation with other kinds of network architecture.

The evaluation metrics presented in Table 3, summarize the final model's performance and reflect the success of the project. They indicate that this model is able to predict activity type successfully and is able to correctly distinguish between similar activities in a satisfactory manner.

The limited number of subjects (n=8) in the dataset could potentially lead to a negative impact the model's effectiveness in real-world scenarios, as it was trained with a dataset that is not representative. A larger and more diverse dataset would likely enhance the model's robustness and its capacity to generalize across a wider range of subjects' inputs, thereby improving its applicability in varied settings.

In summary, this project successfully demonstrated the efficacy of the proposed 1D CNN model for activity recognition in time-series data, emphasizing its suitability to predict types of activities using physiological data gathered from IMUs.

References

[1] A. Reiss, "PAMAP2 Physical Activity Monitoring." UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5NW2H.

[2] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022.

[3] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition.," in *Ijcai*, vol. 15, pp. 3995–4001, Buenos Aires, Argentina, 2015.

[4] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical systems and signal processing*, vol. 151, p. 107398, 2021.