Original Research Paper

Optimization of Multi-Layer Perceptron Deep Neural Networks using Genetic Algorithms for Hand Gesture Recognition

[1]Khanh Nguyen-Trong and [2]Thi-Thanh-Tan Nguyen

[1]*Naver AI Lab, Posts and Telecommunications Institute of Technology, Ha Noi, Viet Nam*
[2]*Faculty of Information Technology, Electric Power University, Ha Noi, Vietnam*

Article history
Received: 03-12-2021
Revised: 19-01-2022
Accepted: 27-01-2022

Corresponding Author:
Khanh Nguyen-Trong
Naver AI Lab, Posts and Telecommunications Institute of Technology, Ha Noi, Viet Nam
Email: khanhnt@ptit.edu.vn

Abstract: Applications of wearable sensors for Hand Gesture Recognition (HGR) have been gaining popularity in recent years. Among the proposed methods, deep neural networks with many hidden layers are promising to address the requirements of this wearable activity recognition. They can directly uncover features tied to the dynamics of HGR, from simple motion encoding in lower layers to more complex motion dynamics in upper layers. However, these methods require many efforts of researches to build an efficient neural network architecture. This study proposes an integrated method that allows finding the best neural networks for HGR using wearable sensors. The proposed method consists of two parts: (i) A generic Multi-Layer Perceptron (MLP) deep neural network and (ii) A genetic algorithm. We applied the genetic algorithm to find the best network architecture in terms of accuracy. At each generation of the algorithm, a new set of architecture was created with different Hyper parameters (the activation, optimizer, the number of layers, neurons and epochs). Extensive experiments were conducted on a dataset containing 18.000 gesture samples from 20 subjects. Experimental results demonstrated the performance and efficiency of the proposed methods in finding deep neural network architectures for HGR. The obtained neural network achieves 89.21% of accuracy and outperforms the previous study on the same dataset.

Keywords: Hand Gesture Recognition, MLP Deep Neural Networks, Genetic

Introduction

Hand gesture is considered to be a mental concept of a human idea associated with an action, response, or a requirement that users realize intending to achieve a result (Pavlovic *et al*., 1997). Hand gesture-based interaction has attracted huge attention from researchers in Human-Computer Interaction (HCI). Previously, many works have focused on computer vision to recognize hand gestures. These approaches usually face challenges related to environmental settings. The recognition performance highly depends on constraints such as lighting conditions, cluttered backgrounds, occlusions and so on. Ultrasonic/optical sensors are also common devices that have been used to capture hand gestures. But, this approach might struggle with practical difficulties in detecting human gestures at any location (Zhang *et al*., 2019a).

The recent development of microelectronic technologies has promoted the proliferation of mobile sensors such as Inertial Measurement Units (IMU), GPS, thermal, vision .... We can easily find IMU sensors in many popular wearable and mobile devices. Thus, they open up many chances for Hand Gesture Recognition (HGR) (Trong *et al*., 2019). Such sensor-based approaches collect sequential data from these sensors and dynamically analyze hand gestures by two main methods: (i) The traditional machine learning and (ii) deep learning.

Many traditional methods can be found in the literature, such as dynamic time warping, k-means clustering, decision trees, support vector machines and so on. However, due to its high accuracy, deep learning gradually replaces the traditional techniques in human activity recognition. Numerous Deep Neural Networks (DNN) have been proposed for analyzing low-level sensing signals to infer high-level human activities and gestures (Zhang *et al*., 2019b). Neural networks such as Multi-Layer Perceptron (MLP) with more than one hidden layer (Tamim *et al*., 2020), Convolution Neural Network (CNN) (V. and R., 2020) network, Recurrent Neuron Network (RNN) (Xing *et al*., 2020), have proven their ability to achieve excellent performance in the field (Pham *et al*., 2020a; Reissner *et al*., 2019; Kwon *et al*., 2018; Leung *et al*., 2018; Lee *et al*., 2017; Hong *et al*., 2016; Iyer *et al*., 2016; Kratz *et al*., 2013).

In general, the sensor data are usually used as input to these networks, directly (without any transformation) or indirectly (transforming to other formats). Then, after being processed by some hidden layers, corresponding hand gestures are predicted at the output layer. In this context, architectural variants of networks highly influence prediction accuracy. However, designing an efficient DNN architecture is a challenging task, which requires much effort from researchers. Several factors should be considered to have an optimized DNN network, such as the number of hidden layers, neurons in each hidden layer, or functions that connecting the neurons. We necessitate a combinatorial search over architectures and their Hyper parameters (Turek *et al*., 2019). Brute force trial and error is a popular solution: Researchers try every combination of sensible parameters and compare the obtained accuracy. But, exploring all options is difficult and expensive, due to it takes a long time to find the most optimized solution.

Genetic Algorithm (GA) is a directed heuristic search technique proposed by (Holland, 1992). Starting from an initial population, the algorithm bases on the processes of mating, breeding and activities such as selection, cross-exchange and mutation, to create new, more optimal individuals (Katoch *et al*., 2021). These processes use an objective function to produce genetic variability. This idea has significant similarities with the problem of DNN optimization. We can apply GA to perform parallel searches in a set of different DNN (population) and toward an optimal solution proceeds by maintaining a population of solutions from which new structures (a new number of hidden layers, of neurons or activations) are created using genetic operators. Therefore, in this study, we propose a GA to evolve and find optimal hyper parameters of neural network architectures for HGR. The remaining of the paper is structured as follows. Section 'Related work' describes related works; Section 'Materials and Methods' details our proposed networks. The experimental evaluation is present in Section 'Experimental Results' and the paper ends up with the conclusion and discussion.

Related Study

In general, a HGR system consists of two main steps, as illustrated in Fig. 1: (i) Pre-processing comprised of 3 sub-steps: Data Cleaning, Data Segmentation and Data Transformation; and (ii) Training/Recognition that contains Feature Extraction and Learning/Inference sub steps.

The input signals of this process can come from different kinds of wrist-worn inertial sensors in certain time intervals. These data are time-dependent, highly fluctuating and oscillatory, which makes them difficult to recognize underlying patterns (Lara and Labrador, 2013; Zheng *et al*., 2018). Therefore, they need to be clean by related techniques at the *Data Cleaning* step, for example, reducing noise, detecting N/A values, detecting gaps, outlier analysis, normalization (Naduvil-Vadukootu *et al*., 2017) and so on. For instance, (Haseeb and Parasuraman, 2019) presented an online machine learning solution for recognizing touch-less hand gestures on a smartphone (mobile device). The authors applied a noise detection model to filter only interested signals, mean subtraction, the re-sampling technique for normalization data raw. (Mezari and Maglogiannis, 2018) also introduced a heuristic algorithm to eliminate tap events that are considered as noise by using a threshold value.

The input time-series data consists of multiple data points in chronological order, where a gesture is repeated across a short interval. Therefore, after data cleaning, we usually segment the continuous input streams in individual gestures with a fixed-length size (Liu *et al*., 2018). Sliding windows is one of the most common techniques using for segmentation, as in the works of (Zhang *et al*., 2019a; Lee and Lee, 2018; Naduvil-Vadukootu *et al*., 2017; Ordo´nez˜ and Roggen, 2016).

Depending on the applied training/recognition method in the next step, the segmented data can be then transformed or not into other forms at the *Data Transformation* step. For example, (Zheng *et al*., 2018), transformed data into four types of images, including raw plots, multichannel plots, spectrogram and a combination of spectrogram and sallow features. (Zhang *et al*., 2018) also used spectrogram as input for the training/recognition. Therefore, the authors transform time-series data into corresponding forms. But, in many other works, the time series segmented data are then passed directly to the next step without transformation. It can be found in the works of (Trong *et al*., 2019; Pancholi and Joshi, 2019; Ordonez and automatically performed by Deep neural network models Roggen, 2016; Haseeb and Parasuraman, 2019; Zhang *et al*., 2019a; Liu *et al*., 2018).

At the Training/Recognition step, in the literature, researchers typically perform two functions: Feature extraction and Learning/inference to produce the recognition model (Training purpose) or predict corresponding activities (Recognition purpose). Regarding the development of deep learning technologies in mining time-series data, we divide methods at this step into two categories: (i) The traditional machine learning, as presented in Fig. 1a; and (ii) the deep learning, as presented in Fig. 1b.

The traditional approach uses hand-crafted features related to the user's movement (e.g., hand gestures), environmental variables (e.g., temperature and humidity), or physiological signals (e.g., heart rate or electrocardiogram). There are three methods to extract these features from time-series data: Statistical, structural and hybrid (Olszewski *et al*., 2001). The first methods extract features from quantitative attributes of data by using, for example, the Fourier or the Wavelet transform. The second one's base on the interrelationship among data. The last ones combine both of them to extract features. These features are passed then to the Learning/Inference step, where a various range of machine learning techniques are used to lean (Training purpose) or to classify (Recognition purpose) the features. They usually base on similarity (e.g., Template matching, k-Nearest neighbor), probability (e.g., Bayes rule), boundaries (e.g., decision trees, neural networks) and clustering (e.g., k-means, hierarchical) methods.

Regarding the second approach, these steps are automatically performed by Deep Neural Networks (DNN), as illustrated in Fig. 1(b). These networks consist of numerous hierarchical layers of non-linear processing units, in which each layer processes the outputs of the previous layer (Trong *et al*., 2019). This architecture allows us to automatically extract features and classify them so that there is no need for manual works. Because of sequential and temporal characteristics of collected data, HGR using wearable sensors is suitable with deep learning methods that have memory structure, such as Recurrent Neural Network (RNN) (Jian *et al*., 2019; Ameur *et al*., 2020).

For instance, (Ordo´nez and Roggen, 2016) proposed a deep learning framework composed of Convolutional Neural Networks (CNNs) and LSTM recurrent layers, that is capable of automatically learning feature representations of hand gestures and modeling the temporal dependencies between their activation. In this study, the authors proposed a network, namely DeepConvLSTM, that contains 8 layers in which the first 5 layers are the input (Layer 1) and the Convolutional ones (layer 2 to 5); the layers 6 and 7 are LSTMs and the last one is a Softmax layer.

Similarly, (Koch *et al*., 2019) presented a stacked recurrent neural network that combined the feature extraction ability of CNNs with LSTM to classify hand gestures. The data of this study come from magnetometer sensors. Unlike to the work of (Ordo´nez˜ and Roggen, 2016), the authors applied the convolutional LSTM (ConvLSTM) (Shi *et al*., 2015) with the standard LSTM in their network.

However, creating efficient deep neural network architectures is a challenging task, which requires much effort from researchers. The exhaustive trial and error approaches are usually conducted to find good architectures, which is time-consuming. In this study, we propose a genetic algorithm to evolve and find such architectures for HGR.

Materials and Methods

Dataset

This study used the GesHome dataset presented in the work of (Nguyen-Trong *et al*., 2021). The dataset contains 18 hand gestures and 6 ongoing gestures (Start to do a gesture and Unknown) from 20 volunteer participants, as illustrated in Fig. 3. We conducted a 5 days' collection period for each participant, in which he/she realized 50 times for each gesture. Thus, we obtained a total of 18000 gesture samples. GesHome contains two groups: The first group consisting of 8 simple gestures and the second group including ten gesture numbers from 0 to 9. It was observed that users were able to remember gestures after only several tries.

Pre-Processing

We used signal collected from an accelerometer and a gyroscope sensor to recognize hand gestures. The raw data is a continuous stream of one-time values. This later contains six-axis values that refer to combinations of three-dimensional data (x, y, z) of the two sensors, as shown in Fig. 2. Therefore, we applied two techniques to normalize and segment the raw data into sequence of separated windows. These windows were then used as the input of the proposed DNN. We used the sliding window technique with 2 sec of the time window and 50% s of overlapping. For each window, the label will be named by the most frequent in 50 raw data respectively.

MLP Deep Neural Network Architecture

In this study, we explored MLP deep neural networks (Severin, 2020) for hand gesture recognition with the input data coming from accelerometer and gyroscope sensors. In general, a Multilayer Perceptron (MLP) is a feed-forward Artificial Neural Network (ANN), which consists of an input layer, a hidden layer and an output layer. A MLP with more than one hidden layer can be considered as DNN (Dey *et al*., 2017; Chinnathambi *et al*., 2018; Bernardi *et al*., 2019; Fallucchi and Cabroni, 2021). In this architecture, every layer contains a bias neuron, except the output layer. They are fully connected to the next layers.

The architecture of such networks depends on the choice of the number of layers, the number of neurons in the hidden layer, the used objective and optimized functions. In this study, we propose a genetic algorithm to find the best MLP deep neural network in term of the accuracy. We tried different architectures and hyper parameters, as follows:

- Number of hidden layers (*nl*): We changed the number of layers, after each generation of the genetic algorithm
- Neurons per hidden layer (*nr*): Two scenarios were applied. First, we randomly generated a number of neurons at each layer. Second, we calculated this number based on Eq 1. It gradually reduced the numbers of neurons from the first layer to the last one

- Activation function: At each generation, we randomly selected an activation for all networks from four functions, including Relu, Elu, Tanh, Sigmoid
- Network optimizer: Similarly, the optimizer is randomly chosen from seven optimizers, including Rmsprop, Adam, Sgd, Adagrad, Adadelta, Adamax, Nadam

Figure 2 details the architecture of networks. As mentioned, we applied the sliding window technique to generate multiple fixed length samples. The window size was set to 2 sec, with an overlap of 1 sec Each window contains 50 samples in X-axis, Y-axis and Z-axis. Thus, the input of networks is a *(50x6)* matrix.

The output layer contains 24 neurons that are corresponding to 24 gestures. Therefore, we specified the neuron number at each hidden layer to gradually reduce toward to 24. Let *nl* the number of hidden layers, the number of neurons *nr* at each hidden layer *i* is determined as follows:

$$nr_i = 16 * w^{nl-i+1}, w \geq 2, nl \geq 2, i \in (1, nl), \tag{1}$$

where, w is a random number that is greater or equal to 2. For example, with $w = 2$ and $nl = 4$ (four hidden layers), the number of neurons at each layer is (from input to output layer) 300, 256, 128, 64, 32 and 24.

Genetic Algorithm

We applied Genetic Algorithms (GAs) (Katoch *et al*., 2021) to find the most optimized neural networks (and the corresponding hyper parameters) for our hand gesture dataset. The algorithms, which are based on the theory of evolution, are widely applied in complex optimization problems. Through genetic operators, GAs explore potential search solutions and escape local optima.

The algorithm is presented as in Algorithm 1. Firstly, we initiate a population that contains nb population networks. Each network has a specific architecture and hyper parameters, which are presented in the previous section.

At each generation, we applied the early stop technique for training and then used accuracy as the fitness function. After each generation, we sort all networks by the accuracy and keep χ percents of the top networks for the next generation and breed children. Lastly, we mutated μ percents of bad networks in terms of accuracy and let the other networks die.

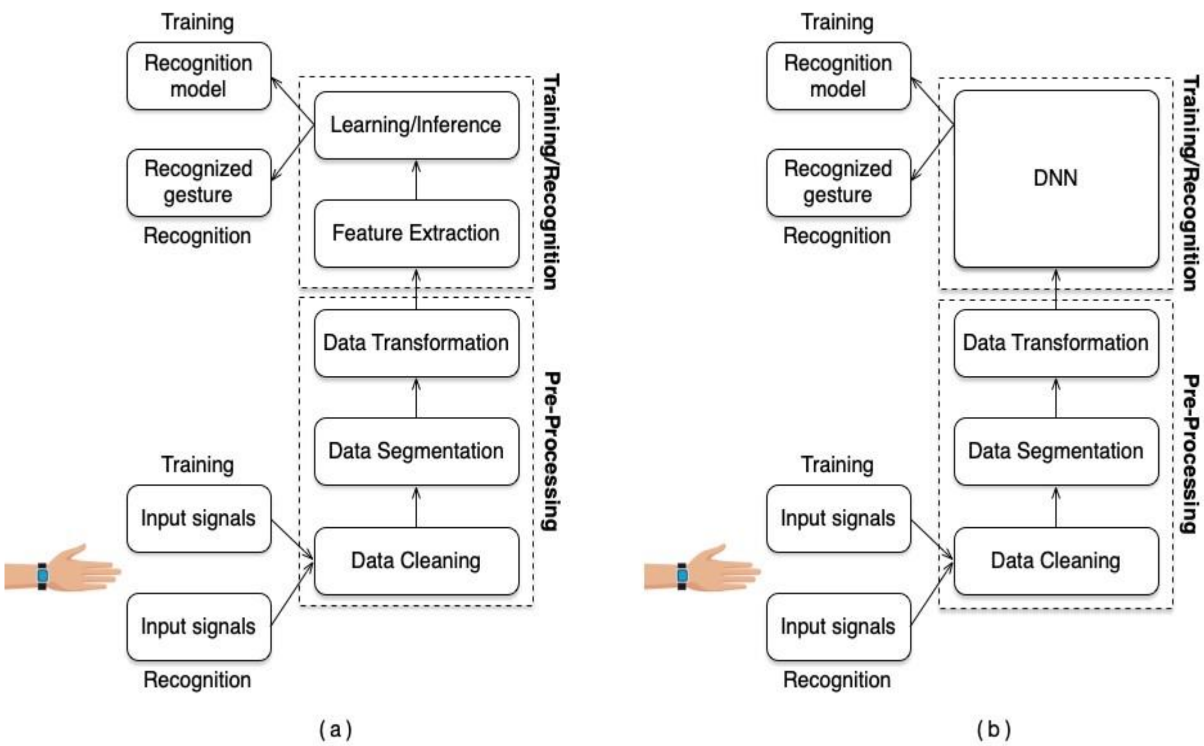


Fig. 1: Method flow for wearable-based hand gesture recognition: (a) General Flow with traditional machine learning approaches; (b) DNN Flow with deep learning approaches where Feature Extraction and Learning and Inference are automatically performed by Deep neural network models

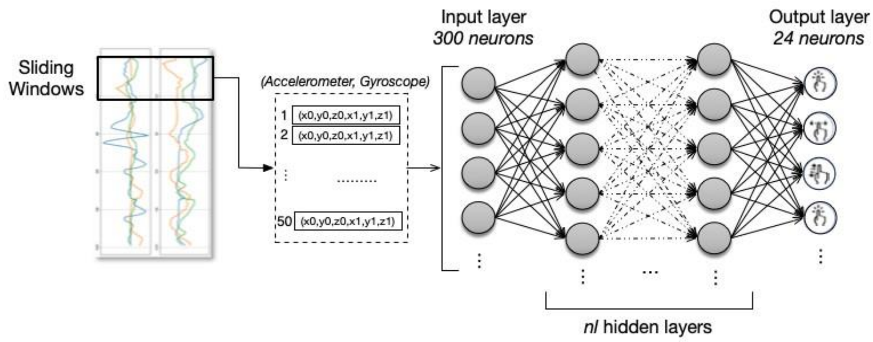


Fig. 2: General architecture of networks

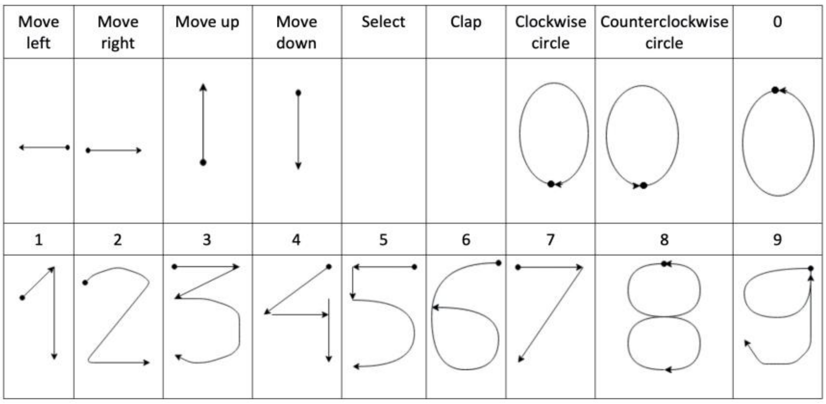


Fig. 3: GesHome dataset (Trong *et al*., 2019)

Experiments

Experiment Setup

We conducted three experiment scenarios, in which each one contains 20 generations, 25 individual networks (populations). Each network has several layers that range from 2 to 10. The maximum epochs of training were set to 5000. We also applied the early stopping technique, with a patience of 1250 epochs. With each experiment, we performed three scenarios with different numbers of neurons, as detailed in Table 1:

- Experiment #1 (Exp1): Each layer had a random number of neurons that ranges from 2 to 4000
- Experiment #2 (Exp2): The layer i has $16{*}w^{nl-i+1}$ neurons, where $w \in [2,3,4]$
- Experiment #3 (Exp3): We added a dropout layer just after each hidden layer of the two previous experiments

All networks were trained using Keras on top of Tensor flow 2.6.0 (Shazeer *et al*., 2018) and Python 3.7, on an NVIDIA Tesla K80 GPU with a 12 GB memory and an Intel (R) 2.3Ghz Xeon(R) microprocessor.

Furthermore, the following techniques and parameters were used to train all networks:

- A categorical cross-entropy function was utilized as the loss function
- An early stop technique was employed to increase the training speed and reduce overfitting. This makes the model stop learning if it has reached its maximum accuracy
- The dataset was imbalanced, in which 80% of data was the 'Start' and 'Unknown' gesture. Therefore, we applied a class weight to make the model pay more attention to samples from an under-represented class. Table 2 details the weight of each class in experiments
- We divided the dataset into three subsets: Training (60%), validation (15% and testing sets (25%). The training and validation set was used to train and valid models, while the last set was used for testing models

Table 1: Experiment parameters

Params	Values
Generation (g)	20
Population (Networks)	25
Number of epochs	Early stopping or 5000 epochs
Number of Layers (nl)	$nl_g \in [2\text{-}10]$

DOI: 10.3844/jcssp.2022.57.66

Table 2: Class weight

Gesture	Weight	Total
0	15	539
1	20	392
2	16	502
3	15	532
4	18	455
5	16	507
6	17	461
7	20	400
8	13	606
9	15	527
CCWCircle	15	579
CWCircle	15	564
Clap	22	382
Move Down	14	591
Move Left	14	578
Move Right	15	539
Move Up	15	547
Select	20	447
Start Gesture	1	7082
Start Move Down	28	316
Start Move Left	30	309
Start Move Right	28	319
Start Move Up	31	265
Unknown	1	7885

Results

After 20 generations, which spent about 24 h for training, we obtained the highest accuracy of 89.21% on the test set. The network belonged to the second experiment, Table 3. Figure 4 details the progress of loss and accuracy on the training and validation set. Owing to the early stop technique, the training was stopped after 1514 epochs. The gap between training loss and validation loss is extremely small, which means that the model operated accurately, without any over-fitting. Table 4 details the F1 score, precision and recall of this network. The network can accurately recognize eight gestures, including ”0” ”1” ”8” ”9” ”CCW Circle”, ”CWCircle”, ”Move Up”, ”Start Gesture”, which have F1-scores of higher than 90%. But, due to the diversity and similarity, the performance is still low for the ”Select” and especially ”Move Down” gesture (69% for ”Move Down” and 77% for ”Select”).

Algorithm 1 Genetic algorithm to find the best MLP deep neural network for hand gesture recognition

Input: *nb generation, nb population, χ, μ*
Params: *nb layers, nb neurons, nb epochs, activations, optimizers*
Output: loss, accuracy, network (activation, optimizer, epochs, layers)
Initialization
get dataset ()
Population creation
for *i* = 0 to *nb population* **do**
net = *random create network* (*params*)
nets. append(*net*)
end for
Generation:
for *i* = 0 to *nb generation* **do**
train (*nets*)
Get scores for each network and save best model
graded = *fitness* (*nets*)
Crossover and breed
babies = *breed* (*male, female*)
Mutate
parents. ext end (*babies*)
end for
Sort the final population, based on the accuracy nets = *sorted*(*nets*)
return *nets* [0]

Table 5 presents the top five best networks among the three experiment scenarios. On average of all obtained networks, the second scenario achieved the highest accuracy (76.62%), while the first one produces the lowest accuracy (45.5%). It can be explained by the fact that networks in the second experiment can learn features in a hierarchical manner. Therefore, the networks predict data better than the others.

Regarding the third experiment, adding a dropout layer after hidden layers didn't improve the network. With the same network architecture using in the first and second experiments, the accuracy on the test set decreased about 4-5%.

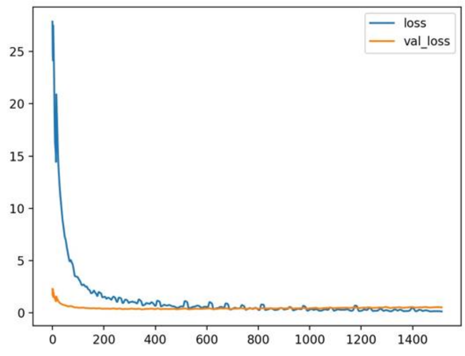
For the activation function and optimizer, *Elu* and *Rmrsprop* seems to produce the best performance. There were 50% networks that used the *Elu* activation function, in the top 10 best networks of the second experiment. The same percentage was observed for the *Rmrsprop* optimizer. The *Sigmoid* activation function produces the worst performance. All networks that used *Sigmoid* function achieved an accuracy of less than 10%.

Discussion

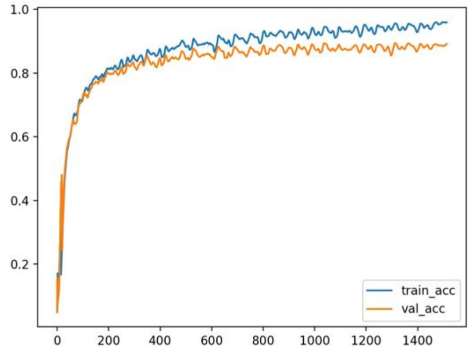
Experimental results show that the obtained network outperforms the previous study on the same dataset (Trong *et al*., 2019) that applied BaselineCNN and DeepConvLSTM, in terms of accuracy (89.21% compared with 73.7and 75.8%). Furthermore, in the previous approach, the authors must use two separated networks: One for detecting starting gestures and another for recognizing the following gestures. In this study, we only need a network to perform both tasks. The results are almost equal with another previous study (Nguyen-Trong *et al*., 2021) that used 1DCNN-BiLSTM (89.21% compared with 90%). But, the obtained network has fewer parameters and layers (249.864 compared with 641.112 parameters, six layers compared with ten layers). Therefore, it results in models with a smaller size and faster inference time. It makes them suitable for running on low-resource devices, such as embed devices, smartphones and so on. However, similarly with the two previous studies, the proposed method did perform well on" Select" and" Move Down" gestures. Besides, due to fully connected architectures of MLP, the training time of our method is longer than the others.

The effectiveness of the proposed solution can also be found in several similar works, such as using GA to design DNN architecture for estimation of pile bearing capacity

(Pham *et al.*, 2020b), or applying MLP to predict risk of diabetes (Fallucchi and Cabroni, 2021).



(a)



(b)

Fig. 4: Progress of loss and accuracy on the training and validation set

Table 3: Optimal network hyper parameter

Parameter	Value
Number of hidden layers	6
Number of Neurons	[300, 288, 240, 192, 144, 96, 48, 24]
Activation	Elu
Optimizer	Rmsprop
Epochs	1514 (Early stopping)
Total parameters	249,864

Table 4: Precision, Recall and F1-Score

Gesture	precision	recall	f1-score
0	0.88	0.92	0.90
1	0.91	0.88	0.90
2	0.91	0.86	0.88
3	0.90	0.86	0.88
4	0.88	0.89	0.89
5	0.89	0.83	0.86
6	0.89	0.88	0.88
7	0.87	0.85	0.86
8	0.90	0.91	0.91
9	0.91	0.88	0.90
CCWCircle	0.95	0.92	0.94
CWCircle	0.94	0.94	0.94
Clap	0.86	0.88	0.87
Move Down	0.61	0.80	0.69
Move Left	0.88	0.86	0.87
Move Right	0.82	0.94	0.88
Move Up	0.93	0.95	0.94
Select	0.94	0.65	0.77
Start Gesture	0.91	0.93	0.92
Start Move Down	0.78	0.82	0.80
Start Move Left	0.90	0.86	0.88
Start Move Right	0.89	0.90	0.89
Start Move Up	0.87	0.89	0.88
Unknown	0.90	0.88	0.89
Average	0.89	0.89	0.89

Table 5: The top five best networks among three experiment scenarios

Order	Scenario	Acc	Layers and Neurons	Activation	Optimizer	Epochs
1	Exp2	89.21%	[288, 240, 192, 144, 96, 48]	Elu	Rmsprop	1514 (Early Stopping)
2	Exp1	88.31%	[2408, 3724, 1175, 1182,2543, 325, 3605, 3795, 1445]	Tanh	Adamax	1892 (Early Stopping)
3	Exp2	88.18%	[288, 240, 192, 144, 96, 48]	Elu	Adam	1552 (Early Stopping)
4	Exp1	88.12%	[1229, 132, 2230, 2431] 325, 3605, 3795, 1445]	Relu	Adam	1317 (Early Stopping)
5	Exp2	87.70%	[288, 240, 192, 144, 96, 48]	Relu	Rmsprop	1775 (Early Stopping)

Conclusion

A method for recognizing hand gestures has been proposed in this study, in which we employed data coming from popular sensors embedded inside wearable devices. We proposed a general MLP deep neural network for analyzing, learning features from sensing signals. Then, a genetic algorithm was applied to find the most efficient network architecture. At each generation of the algorithm, several hyper parameters were tried for better network architecture, including the activation function, optimizer, the number of layers, of neurons. After 20 generations, we obtained the best MLP deep neural network with an accuracy of 89.21% on the testing set, which outperforms previous studies on the same dataset. In the future, we will extend the dataset to add more gesture, as well as to balance different classes. Moreover, the genetic architecture will be applied to find optimal architecture of other DNNs, such as CNN, LSTM and so on.

Funding Information

This study was partly support by the Ministry of Information and Communications under the grant no. DT.028/19.

Author’s Contributions

Khanh Nguyen-Trong: Designed the research plan and organized the study; coordinated the data-analysis and contributed to the writing of the manuscript; implemented the proposed method and performed all the experiment.

Thi-Thanh-Tan Nguyen: Participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

Ethics

This article is unique and contains unpublished material. The comparing creator affirms that all of different writers have perused and endorsed the composition what’s more no moral issues included.

References

Ameur, S., Ben Khalifa, A., & Bouhlel, M. S. (2020). A novel hybrid bidirectional unidirectional lstm network for dynamic hand gesture recognition with leap motion. Entertainment Computing, 35, 100373. doi.org/10.1016/j.entcom.2020.100373

Bernardi, M. L., Cimitile, M., Martinelli, F., & Mercaldo, F. (2019). Keystroke analysis for user identification using deep neural networks. In 2019 International Joint Conference on Neural Networks (IJCNN), 1-8. doi.org/10.1109/IJCNN.2019.8852068

Chinnathambi, R. A., Plathottam, S. J., Hossen, T., Nair, A. S., & Ranganathan, P. (2018). Deep neural networks (dnn) for day-ahead electricity price markets. In 2018 IEEE Electrical Power and Energy Conference (EPEC), 1–6. doi.org/10.1109/EPEC.2018.8598327

Dey, P., Ghosh, A., & Pal, T. (2017). Regularized stacked auto-encoder based pre-training for generalization of multi-layer perceptron. In Mart´ın-Vide, C., Neruda, R. and Vega-Rodr´ıguez, M. A., editors, Theory and Practice of Natural Computing, 232–242, Cha Springer International Publishing. doi.org/10.1007/978-3-319-71069-3_18

Fallucchi, F., & Cabroni, A. (2021). Predicting Risk of Diabetes using a Model based on Multilayer Perceptron and Features Extraction. Journal of Computer Science, 17(9), 748–761. doi.org/10.3844/JCSSP.2021.748.761

Haseeb, M. A. A. & Parasuraman, R. (2019). Wisture: Touch-less hand gesture classification in unmodified smartphones using wi-fi signals. IEEE Sensors Journal, 19(1) 257–267. doi.org/10.1109/JSEN.2018. 2876448

Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence. MIT press.

Hong, F., You, S., Wei, M., Zhang, Y., & Guo, Z. (2016). MGRA: Motion gesture recognition via accelerometer.

Iyer, D., Mohammad, F., Guo, Y., Al Safadi, E., Smiley, B. J., Liang, Z., & Jain, N. K. (2016). Generalized hand gesture recognition for wearable devices in IoT: Application and implementation challenges. In Perner, P., editor, Machine Learning and Data Mining in Pattern Recognition, 346–355, Cham. Springer International Publishing. doi.org/10.1007/978-3-319-41920-6_26

DOI: 10.3844/jcssp.2022.57.66

Jian, C., Li, J., & Zhang, M. (2019). Lstm-based dynamic probability continuous hand gesture trajectory recognition. IET Image Processing, 13(12), 2314–2320. doi.org/10.1049/iet-ipr.2019.0650

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present and future. Multimedia Tools and Applications, 1 – 36. doi.org/10.1007/s11042-020-10139-6

Koch, P., Dreier, M., Bohme, M., Maass, M., Phan, H., & Mertins, A. (2019). Inhomogeneously stacked rnn for recognizing hand gestures from magnetometer data. In 2019 27th European Signal Processing Conference (EUSIPCO), 1-5. doi.org/10.23919/EUSIPCO.2019.8903132

Kratz, S., Rohs, M., & Essl, G. (2013). Combining acceleration and gyroscope data for motion gesture recognition using classifiers with dimensionality constraints. In Proceedings of the 2013 International Conference on Intelligent User Interfaces, IUI ’13, 173–178, New York, NY, USA. ACM. doi.org/10.1145/2449396.2449419.

Kwon, M. C., Park, G., & Choi, S. (2018). Smartwatch user interface implementation using CNN-based gesture pattern recognition. Sensors (Switzerland), 18(9) 1–12. doi.org/10.3390/s18092997

Lara, O. D. & Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. IEEE Communications Surveys Tutorials, 15(3) 1192-1209. doi.org/10.1109/SURV.2012. 110112.00192

Lee, B. G. & Lee, S. M. (2018). Smart wearable hand device for sign language interpretation system with sensors fusion. IEEE Sensors Journal, 18(3) 1224-1232. doi.org/10.1109/JSEN.2017.2779466

Lee, W.-H., Liu, X., Shen, Y., Jin, H., & Lee, R. B. (2017). Secure Pick Up: Implicit Authentication When You Start Using the Smartphone. doi.org/10.1145/3078861.3078870

Leung, H.-M. C., Fu, C.-W., &Heng, P.-A. (2018). Twistin: Tangible authentication of smart devices via motion co-analysis with a smartwatch. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 2(2),72:1-72:24. doi.org/10.1145/3214275

Liu, Y., Zhang, Y., & Zeng, M. (2018). Novel algorithm for hand gesture recognition utilizing a wrist-worn inertial sensor. IEEE Sensors Journal, 18(24), 10085-10095. doi.org/10.1109/JSEN.2018.2873003

Mezari, A. & Maglogiannis, I. (2018). An easily customized gesture recognizer for assisted living using commodity mobile devices. Journal of Healthcare Engineering, 2018, 2040–2295. doi.org/10.1155/2018/3180652

Naduvil-Vadukootu, S., Angryk, R. A., & Riley, P. (2017). Evaluating preprocessing strategies for time series prediction using deep learning architectures. In FLAIRS 2017 - Proceedings of the 30th International Florida Artificial Intelligence Research Society Conference, pages 520-525.

Nguyen-Trong, K., Vu, H. N., Trung, N. N., & Pham, C. (2021). Gesture recognition using wearable sensors with bi-long short-term memory convolutional neural networks. IEEE Sensors Journal, 21(13), 15065-15079. doi.org/10.1109/JSEN.2021.3074642

Olszewski, R. T., Maxion, R. A., & Siewiorek, D. P. (2001). Generalized feature extraction for structural pattern recognition in time-series data. PhD thesis, School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213.

Ordo´nez, F. J. & Roggen, D. (2016). Deep convolutional˜ and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, 16(1). doi.org/10.3390/s16010115

Pancholi, S. & Joshi, A. M. (2019). Electromyographybased hand gesture recognition system for upper limb amputees. IEEE Sensors Letters, 3(3), 1–4. doi.org/10.1109/LSENS.2019.2898257

Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7),677-695. doi.org/10.1109/34.598226

Pham, C., Nguyen-Thai, S., Tran-Quang, H., Tran, S., Vu, H., Tran, T., & Le, T. (2020a). Senscapsnet: Deep neural network for non-obtrusive sensing based human activity recognition. IEEE Access, 8, 86934–86946. doi.org/10.1109/ACCESS.2020.2991731

Pham, T. A., Tran, V. Q., Vu, H.-L. T., & Ly, H.-B. (2020b). Design deep neural network architecture using a genetic algorithm for estimation of pile bearing capacity. PLOS ONE, 15(12), 1-25. doi.org/10.1371/journal.pone.0243030

Reissner, L., Fischer, G., List, R., Giovanoli, P., & Calcagni, M. (2019). Assessment of hand function during activities of daily living using motion tracking cameras: A systematic review. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, 233(8), 764-783. doi.org/10.1177/0954411919851302

Severin, I.-C. (2020). Time series feature extraction for head gesture recognition: Considerations toward hci applications. In 2020 24th International Conference on System Theory, Control and Computing (ICSTCC), 232-237. doi.org/10.1109/ICSTCC50638.2020.9259741

Shazeer, N., Cheng, Y., Parmar, N., Tran, D., Vaswani, A., Koanantakool, P., Hawkins, P., Lee, H., Hong, M., Young, C., Sepassi, R., & Hechtman, B. (2018). Meshtensorflow: Deep learning for supercomputers.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., & Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS15, 802–810, Cambridge, MA, USA. MIT Press. doi.org/10.5555/2969239.2969329

Tamim, N., Elshrkawey, M., Abdel Azim, G., & Nassar, H. (2020). Retinal blood vessel segmentation using hybrid features and multi-layer perceptron neural networks. Symmetry, 12(6). doi.org/10.3390/sym12060894

Trong, K. N., Bui, H., & Pham, C. (2019). Recognizing hand gestures for controlling home appliances with mobile sensors. In 2019 11th International Conference on Knowledge and Systems Engineering (KSE), 1-7. doi.org/10.1109/KSE.2019.8919419

Turek, J. S., Jain, S., Capota, M., Huth, A. G., & Willke, T. L. (2019). A single-layer rnn can approximate stacked and bidirectional rnns and topologies in between. CoRR, abs/1909.00021. doi.org/10.1109/JSEN.2018.2876448

Xing, Y., Di Caterina, G., & Soraghan, J. (2020). A new spiking convolutional recurrent neural network (scrnn) with applications to event-based hand gesture recognition. Frontiers in Neuroscience, 14, 1143. doi.org/10.3389/fnins.2020.590164

Zhang, X., Yang, Z., Chen, T., Chen, D., & Huang, M. (2019a). Cooperative sensing and wearable computing for sequential hand gesture recognition. IEEE Sensors Journal, 19(14), 5775–5783. doi.org/10.1109/JSEN.2019.2904595

Zhang, Z., Tian, Z., & Zhou, M. (2018). Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor. IEEE Sensors Journal, 18, 3278-3289. doi.org/10.1109/jsen.2018.2808688

Zhang, Z., Tian, Z., Zhang, Y., Zhou, M., & Wang, B. (2019b). u-deephand: Fmcw radar-based unsupervised hand gesture feature learning using deep convolutional auto-encoder network. IEEE Sensors Journal, 19(16), 6811–6821. doi.org/10.1109/JSEN. 2019.2910810

Zheng, X., Wang, M., & Ordieres-Mere, J. (2018). Comparison of data preprocessing approaches for applying deep learning to human activity recognition in the context of industry 4.0. Sensors, 18(7). doi.org/10.3390/s18072146