WALL-FOLLOWING ROBOT NAVIGATION CLASSIFICATION USING DEEP LEARNING WITH SPARSE CATEGORICAL CROSSENTROPY LOSS FUNCTION

Ahnaf Ramadhan Angga Raksa
Faizar Padukawan
Kahfi Kurnia Aji
Muhammad Rizki Alamsyah
Shelly Octaviyani
Eka Angga Laksana

DOI: https://doi.org/10.37178/ca-c.23.1.412

Ahnaf Ramadhan Angga Raksa, *Department of Informatics Engineering, Faculty of Engineering Widyatama University, Bandung, Indonesia*
Email: ahnaf.ramadhan@widyatama.ac.id

Faizar Padukawan, *Department of Informatics Engineering, Faculty of Engineering Widyatama University, Bandung, Indonesia*
Email: faizar.padukawan@widyatama.ac.id

Kahfi Kurnia Aji, *Department of Informatics Engineering, Faculty of Engineering Widyatama University, Bandung, Indonesia*
Email: kahfi.kurnia@widyatama.ac.id

Muhammad Rizki Alamsyah, *Department of Informatics Engineering, Faculty of Engineering Widyatama University, Bandung, Indonesia*
Email: m.rizki@widyatama.ac.id

Shelly Octaviyani, *Department of Informatics Engineering, Faculty of Engineering Widyatama University, Bandung, Indonesia*
Email: shelly.octaviyani@widyatama.ac.id

Eka Angga Laksana, *Department of Informatics Engineering, Faculty of Engineering Widyatama University, Bandung, Indonesia*
Email: eka.angga@widyatama.ac.id

Abstract

Nowadays, technology has developed in advance. One example of things that are experiencing technological advances are robots. Robots are mechanical devices that were created to replace some of the repetitive human jobs. Where can carry out certain tasks either automatically or by human control or programs that are given based on logic. One example is the navigation on the wall following robot, this robot is a robot that can move independently by detecting the "wall" using the sensors on the robot without hitting the “wall”. One of the studies used a wall-following SCITOS-G5 robot *which was installed with 24 ultrasound sensors and it generates a dataset. The dataset can be used as an analysis in this research. Not only robots that experience the development of the times, but the world of research is also experiencing developments, one example is deep learning. This research uses a deep learning neural network method with a keras library in python. The process of this research is to calculate how much accuracy is generated from the deep learning method. The accuracy resulting from this study will be compared with the accuracy in the K-Nearest Neighbor study with the same dataset. The results of the calculation of accuracy in previous studies using the machine learning method K-Nearest Neighbor is 88.17%. Meanwhile, in this study, the accuracy obtained using the deep learning method is 95.27%. Therefore, the use of the deep learning method using a keras library in this study is better than the K-Nearest Neighbor method.*

Keywords: Wall-Following Robot Navigation, Deep Learning, Keras, Python, Sparse Categorical Crossentropy.

Introduction

Artificial intelligence is developing rapidly for the past few years. With artificial intelligence, problems faced by humans that were previously difficult can now be solved easily. One of the artificial intelligences is the concept of deep learning which uses simple representations but with these concept computers can build complex concepts.

Wall Follower is a robot that moves based on obstacles detected by ultrasound sensors. Usually, a simple wall follower robot uses three ultrasound sensors, that is left, right and front of the robot. With these 3 sensors, the robot is able to move forward, turn and backward. To be able to move forward, the sensors on the robot do not detect any walls in all directions, and if the front sensor and the left sensor detect an obstacle, the robot will turn right until the front sensor does not detect an obstacle. Then the movement will be the opposite for sensors on the other side, that's the simple movement of a wall follower robot based on three sensors.

By adding more sensors on each side of the robot, the accuracy of its movement is higher, as is the case with the dataset we use based on the SCITOS G5 robot which has 24 ultrasound sensors. This dataset was obtained from the UCI Machine Learning Repository managed by Ananda Freire, Marcus Veloso and Guilherme Barreto Department of Teleinformatics Engineering at Federal University of Ceará. It can be seen that the data obtained from 24 ultrasound sensors with sensor readings sampled at a speed of 9 samples per second will have differences in their movement. By the movement in question is Move-Forward, Sharp-Right-Turn, Slight-Left-Turn and Slight-Right-Turn.

Based on the dataset that has been obtained, this research will measure the accuracy of the Wall-Following Robot Navigation dataset using a deep learning neural network model method with a keras library in Python. This research will also compare the level of accuracy of the deep learning neural network method with the keras library in Python against the K-Nearest Neighbor algorithm[1].

Literature Review

[2] perform classification by comparing CNN-1D, CNN-2D, SVC, MLR, ANN methods to understand hidden patterns in navigation work and classify actions by robots in terms of different movements performed by robots in response to them. In research [3]using five variations of backpropagation training, namely, gradient descent backpropagation, gradient descent with backpropagation momentum, gradient descent with adaptive learning rate backpropagation, gradient descent with momentum and adaptive learning rate backpropagation and Levenberg marquardt backpropagation. measure the performance of the training and the resulting weights. Then it will be compared based on the accuracy and time that has been obtained from

the training so that it can be determined the most effective variation of backpropagation training for wall-following robot navigation data. [4]conducted research on the state of the art pattern recognition SVM classification using the 1V1 strategy. Optimizing the application of the AI method in the wall-following robot navigation technique to increase the efficiency of its accuracy and computational complexity so as to increase the robot's intelligence in the wall-following robot navigation technique, a multi-class SVM model was obtained using the 1V1 strategy for classifying wall-following robot navigation data with high accuracy greater than MLP (97.59%). [5-7] conducted research using the classification method by creating a new base of objects with unknown classifications. The main objective of this work is to assess the effectiveness of the selected classification with a simple case study, namely a wall following robot. Statistical data analysis was also used in this study. Because of the exploration and types of patterns found in the data mining process, it can be divided intoseveral main classes, such as:as: association, classification, grouping, detection of changes and deviations, or finding sequence patterns. In research[8]focuses on the field of robot navigation where the robot navigation trail is a collection of data that is used to test several classification algorithms using WEKA. The results show some limitations in terms of testing/running time in the KNN algorithm. Set reduction: Nearest Neighbor (CNN) algorithm. NS The KNN algorithm was tested again to see significant improvements in test time for three files and accuracy in one file. So, this time it is able to speed up the KNN algorithm 19 times more with a performance increase rate of more than 100%. In the multi-attribute file the dataset of 4 sensors and 24 sensors has been preprocessed again to remove redundant attributes. Both accuracy and test time are improved using the WEKA attribute selection feature.

Materials and Methods

Wall-Following Robot Navigation Dataset

This dataset was obtained from research by Ananda L. Freire, Guilherme A. Barreto, Marcus Veloso and Antonio T. Varela from the Department of Teleinformatics Engineering, Federal University of Ceará with the research title Short-Term Memory Mechanisms in Neural Network Learning of Robot Navigation Task: A Case Study. The dataset used is taken from the movement of the Wall-following robot, namely SCITOS-G5. There are 24 ultrasound sensors mounted in a circle on the robot starting from the front of the robot and increasing clockwise. Data is taken from the 24 sensors where the SCITOS-G5 moves for 4 turns through the room following the wall in a clockwise direction.[9] This dataset contains 5456 number of instances and 24 number of attributes taken from sensor readings at a rate of 9 samples per second[9].

Deep Learning

Deep learning is a subset of machine learning and artificial intelligence that is used to provide precision tasks such as object detection, language translation, speech recognition, etc. Deep learning is the development of a multiple layer neural network that differs from traditional machine learning techniques by automatically analyzing data such as text, images or videos. Deep learning has the concept of learning working at each different layer and getting representations and abstractions so that the data can be understood[10].

K-Nearest Neighbor

K-Nearest Neighbor (KNN) is an algorithm used to classify an object, based on k training data that are closest to the object. The condition for the value of k is that it cannot be greater than the number of training data, and the value of k must be odd and more than one. Besides being used for classification, the K-NN algorithm is also used for estimation and prediction. The steps of the K-NN algorithm are:

a. Specifies the parameter k (number of nearest neighbors)

b. Calculates the distance (similarity) between all training records and new objects
c. Sort data by distance value from smallest to largest value
d. Retrieve data from a number of k values. Determines the label that occurs most frequently in the k training records closest to the object[11, 12].

Convolutional Neural Network

Convolutional neural network or commonly abbreviated as CNN is one type of deep learning algorithm used in processing input data in the form of images. Convolution is a matrix that functions to filter an image, to "learn" to recognize and distinguish between one image and another. CNN architecture consists of input layer, output layer and hidden layer. In the hidden layer there is a convolutional layer, pooling layer, normalization layer, activation layer, fully connected layer, and loss layer.

Keras

Keras is an API for deep learning methods that use the python programming language. Keras runs on one of the backend platforms, namely Tensorflow. Because Keras is included in deep learning research, Keras can be used to find accuracy in a dataset that has a large amount of data. In addition, Keras also supports computing on the Convolutional Neural Network that can run well on the CPU and GPU. According to the official website it is compatible with python version 2.7 to version 3.6.

Implementation and Results

Algorithm testing was carried out on the wall-following robot dataset which had a number of instances of 5456 and a number of attributes of 24. The number of attributes was used as the input variable (X) and there was one column named “movement” as the output variable (y). Input variable (X):

Table 1

Ultrasound Sensor Angle

Variable	Information
US1	Ultrasound sensor in front of the robot with an angle 180˚
US2	Ultrasound reading with an angle of -165˚
US3	Ultrasound reading with an angle of -150˚
US4	Ultrasound reading with an angle of -135˚
US5	Ultrasound reading with an angle of -120˚
US6	Ultrasound reading with an angle of -105˚
US7	Ultrasound reading with an angle of -90˚
US8	Ultrasound reading with an angle of -75˚
US9	Ultrasound reading with an angle of -60˚
US10	Ultrasound reading with an angle of -45˚
US11	Ultrasound reading with an angle of -30˚
US12	Ultrasound reading with an angle of -15˚
US13	Ultrasound sensor at the back of the robot with angle 0˚
US14	Ultrasound reading with an angle of 15˚
US15	Ultrasound reading with an angle of 30˚
US16	Ultrasound reading with an angle of 45˚
US17	Ultrasound reading with an angle of 60˚
US18	Ultrasound reading with an angle of 75˚
US19	Ultrasound reading with an angle of 90˚
US20	Ultrasound reading with an angle of 105˚
US21	Ultrasound reading with an angle of 120˚
US22	Ultrasound reading with an angle of 135˚
US23	Ultrasound reading with an angle of 150˚
US24	Ultrasound reading with an angle of 165˚

Output variable (y):

1. Slight-Right-Turn
2. Sharp-Right-Turn
3. Move-Forward
4. Slight-Left-Turn

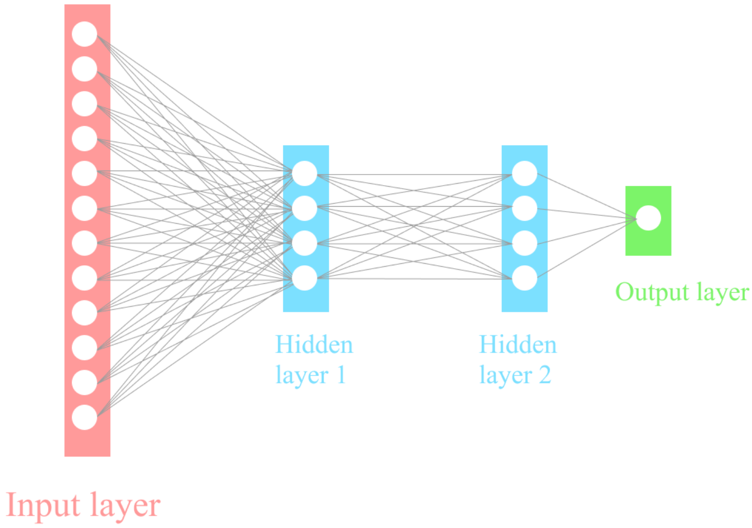

Fig. 1 Neural Network Architectures

The architecture above is a fully-connected layer which has 4 layers, each layer has a certain number of nodes. These nodes are also known as neurons, the neurons are interconnected marked by lines in the architecture above. The first layer is the input layer which has 24 input_dim and 12 nodes. Then the second and third layers are hidden layers where each layer has 4 nodes with certain activation functions. And on the last layer is the output layer.

The model on the keras is defined as a sequence of layers. Layers are added one by one to create a sequential model. The input layer has the right number of input features by determining the number of input variables, which is 24. This research uses a fully connected network structure with three layers. Each layer is defined by the number of nodes and the activation function.

$$f(x) = \max(0, x), \tag{1}$$

The formula above is ReLU or Rectified Linear Unit which is one of the activation functions. ReLU basically creates a limit on the number zero, meaning x then x = 0 and if x > 0 then x = x. ReLU is used in the first hidden layer with four nodes.

$$f(x) = 1/(1 + e\hat{}(-x)). \tag{2}$$

The formula above is a sigmoid function which changes the value of x to be non-linear and has a value of 0 to 1. This activation function is used in layers after layers that use the ReLU activation function with the number of nodes equal to the number of nodes before which is four nodes.

After the model is defined, the model compilation can be done. Backend efficient numerical libraries under the covers such as TensorFlow or Theano are used to compile the model. The backend will automatically determine the best way to represent which network will be used as training to run. In this study, entropy is used as a loss argument where the entropy used is for data classification with output labels of more than 2 categories or non-binary. In keras this loss is defined as "sparse_categorical_crossentropy".

In keras, there are several types of loss function for compiling model. There are three classifications that are related to the output variables used namely binary crossentropy, categorical crossentropy and sparse categorical crossentropy. The difference between the three is to look at the variable output. Binary crossentropy will calculate with two output variables, for example the numbers 1 and 0 or true and false. Sparse categorical crossentropy calculte with two or more output variables, for example the labels have [0, 1, 2] to be used.

So, this research absolutely will use Sparse Categorical Crossentropy because the labels or the outpur varibles have more than two type, there are Move-Forward, Slight-Right-Turn, Sharp-Right-Turn, Slight-Left-Turn. But there is the weakness of this function that the labels should be numerical. For this research, the labels or the output variables must be changed to be numerical. So, each movement will represent the number 0 to 3 that is Slight-Right-Turn represent number 0, Sharp-Right-Turn represent 1, Move-Forward represent 2, Slight-Left-Turn represent 3.

The optimizer will be defined as the efficient stochastic gradient descent algorithm "adam". It was chosen because it can give good results for a wide range of problems. Metrics is defined as accuracy because this research leads to the calculation of accuracy. The training process will be carried out iteratively based on predetermined epochs and batches. In this research, 150 iterations will be carried out, which means that the epochs will be defined as 150 and the batch size as 10. The model will be evaluated using the evaluate function to be passed to input(X) and output(y).

After the code is running, the iteration will run and the accuracy will increase up to the 150th iteration. In the first iteration in epoch 1/150 the accuracy is 0.4653 (46.53%). The accuracy has increased until the last iteration in epoch 150/150, the final result is 0.9527 (95.27%)

```
Epoch 142/150
546/546 [==============================] - 1s 1ms/step - loss: 0.1579 - accuracy: 0.9457
Epoch 143/150
546/546 [==============================] - 1s 1ms/step - loss: 0.1642 - accuracy: 0.9443
Epoch 144/150
546/546 [==============================] - 1s 1ms/step - loss: 0.1586 - accuracy: 0.9446
Epoch 145/150
546/546 [==============================] - 1s 1ms/step - loss: 0.1653 - accuracy: 0.9413
Epoch 146/150
546/546 [==============================] - 1s 1ms/step - loss: 0.1668 - accuracy: 0.9437
Epoch 147/150
546/546 [==============================] - 1s 1ms/step - loss: 0.1651 - accuracy: 0.9417
Epoch 148/150
546/546 [==============================] - 1s 938us/step - loss: 0.1633 - accuracy: 0.9445
Epoch 149/150
546/546 [==============================] - 1s 960us/step - loss: 0.1593 - accuracy: 0.9461
Epoch 150/150
546/546 [==============================] - 1s 1ms/step - loss: 0.1637 - accuracy: 0.9457
171/171 [==============================] - 0s 741us/step - loss: 0.1424 - accuracy: 0.9527
Accuracy: 95.27
```

Fig. 2 Iteration proces of runnig code and the result

Compare this research result with research where the study used the K-NN method to calculate accuracy. The difference in the following table down below.

Table 2

Comparison Analysis

Data set	Evaluation Criteria	K-Nearest Neighbor	Neural Network with Keras
24 sensors file	Number of Instances	5456	5456
24 sensors file	Accuracy(percent)	88.17	95.27

This table presnts the results of calculating the accuracy of two different methods on the same dataset. It can be seen that by using the Neural Network method with the keras library there is an improvement in accuracy when compared to the K-Nearest Neighbor method.

Conclusion

The code that is run to calculate the accuracy of the wall-following robot navigation dataset can be said to be successful. Iteration when the code is run runs smoothly as many as 150 iterations. And obtained the results of an accuracy of 95.27% using the deep learning neural network method with a keras library in python. Then the accuracy results are compared with previous studies using the machine learning method of the K-Nearest Neighbor algorithm. The accuracy in this study was 88.17%. Therefore, the deep learning neural network method is more effective in its implementation to seek accuracy. And also, with the Keras library in python, it shows that there is an improvement in finding accuracy compared to research using the K-Nearest Neighbor algorithm.

References

1. Thanh, N.N., H.D. Hiep, and P.H. Tung, *RELATIONSHIP BETWEEN CHARACTERISTICS AND UNIVERSITY INTENTION: A CASE STUDY OF STUDENTS IN VIETNAM.* PalArch's Journal of Archaeology of Egypt/Egyptology, 2021. **18**(18): p. 44-69.
2. Singh, S.K., *Data -Enhanced Convolution Neural Networks for Wall Following Robot Navigation. International Journal of Engineering and Advanced Technology (IJEAT), 9(1), 4426-4430.* 2019 DOI: https://doi.org/10.35940/ijeat.A1361.109119.
3. Utami, A.T.W. and B.S.S. Ulama, *Application of Backpropagation to Improve Time Effectiveness and Accuracy on Wall-Following Robot Navigation Data.* ITS Jurnal Journal of Science and Arts, 2016. **4**(2).
4. Azis, A.I.S., V. Suhartono, and H. Himawan, *Multi-Class SVM model using 1V1 strategy for Wall-Following Robot Navigation Data Classification.* My Cyber Journal, 2017. **13**(2): p. 8-8.
5. Zdrodowska, M., A. Dardzińska, and A. Kasperczuk. *Using Data Mining Tools in Wall-Following Robot Navigation Data Set*. IEEE DOI: https://doi.org/10.1109/MSM49833.2020.9201730.
6. Visaltanachoti, C., C. Viriyavejakul, and T. Ratanaolarn, *DEVELOPMENT GUIDELINES FOR AN ARTIFICIAL INTELLIGENCE ALGORITHMIC MODEL IN TEACHING FOREIGN.* PalArch's Journal of Archaeology of Egypt/Egyptology, 2021. **18**(18): p. 371-380.
7. Nazir, N., et al., *AN IN-DEPTH ANALYSIS OF APPLICABILITY AND EFFECTIVENESS OF LEARNER-CENTERED APPROACH AT UNDERGRADUATE LEVEL IN PAKISTANI CLASSROOM CONTEXT.* PalArch's Journal of Archaeology of Egypt/Egyptology, 2021. **18**(18): p. 353-370.
8. Madi, S. and R. Baba-Ali. *Classification techniques for wall-following robot navigation: a comparative study*. Springer DOI: https://doi.org/10.1007/978-3-319-99010-1_9.
9. Freire, A.L., et al. *Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study*. IEEE DOI: https://doi.org/10.1109/LARS.2009.5418323.
10. Deng, L. and D. Yu, *Deep learning: methods and applications.* Foundations and trends in signal processing, 2014. **7**(3–4): p. 197-387 DOI: https://doi.org/10.1561/2000000039.
11. Peterson, L.E., *K-nearest neighbor.* Scholarpedia, 2009. **4**(2): p. 1883 DOI: https://doi.org/10.4249/scholarpedia.1883.

12. Younas, F. and Z. Mubeen, *BODY IMAGE SATISFACTION AND EATING ATTITUDES: ROLE OF SOCIODEMOGRAPHIC CHARACTERISTICS.* PalArch's Journal of Archaeology of Egypt/Egyptology, 2021. **18**(18): p. 273-288.