ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE
Paper ID #37971

Introducing Deep Learning on Edge Devices Using A Line Follower Robot

Bassam Alshammari

Bassam has received his bachelor's degree in Mechatronics Engineering Technology at Purdue University Calumet, Indiana. He is currently pursuing his master's degree at Pittsburg State University in Kansas. He is an instructor at Technical and Vocational Training Corporation in Saudi Arabia.

Erik A. Mayer (Professor)

Erik Mayer is a Professor at Pittsburg State University in Kansas where he has been instrumental in forming the Computer and Embedded Systems emphasis in the Electronics Engineering Technology program. His research interests are embedded systems, artificial intelligence, renewable energy, and power electronics. He previously taught at Bowling Green State University in Ohio where he worked with the Electric Vehicle Institute . In addition, he worked at Visteon Corporation designing components for hybrid vehicles. He received his Ph.D. in Engineering Science at the University of Toledo in Ohio.

Zachariah D Woods

Zachariah Woods is an Undergraduate at Pittsburg State University majoring in Electrical Engineering Technology with a minor in Automation and an emphasis in Computer and Embedded Systems. He is the president of the IEE at Pittsburg State and, he has professional interest in automation, artificial intelligence, embedded systems, and renewable energy.

Austin Taylor Smith

Estevan Hernandez

Estevan Hernandez is a student at Pittsburg State University with a major in Electronics Engineering Technology. With an emphasis in Embedded Systems, a minor in Automation, and a minor Mathematics. In addition, he is also a member of Phi Kappa Phi and the National society of leaderships and success.

Kevin Joseph Birk (Student)

Kevin Birk is a student at Pittsburg State University. He is majoring in Electronics Engineering Technology, has a minor in Automation Controls and an emphasis in Computer & Embedded Systems. Kevin is the president of PSU Combat Robotics, the treasurer of PSU Ham Radio club, and is a team lead for an extracurricular project, The Great Lunar Expedition for Everyone. He is looking to graduate in May of 2023.

Trenton Drake Allison

My name is Trenton Allison. I am from Fort Scott Kansas. I am the oldest of 3 other siblings. My grandfather was an electrician and inspired me to become an electrical engineer. I am currently studying at Pittsburg State University to

obtain the Electrical Engineering Technology Degree with a major in Automation. I am a member of the Pitt State combat robotics club as the secretary.

Jacob D Brennon

Jacob Brennon is a student at Pittsburg State University in Kansas where he is pursuing his Bachelor of Science Degree with a major in Electronics Engineering Technology and an emphasis in Automation. He is the secretary of the student chapters of IEEE and ISA at Pittsburg State University.

Caleb Chase

Caleb Chase graduated Pittsburg State University in May 2022 with a Bachelor in Electronics Engineering Technology. His emphasis was in computer and embedded systems, and he has a minor in Computing. He worked closely with Dr. Erik Mayer in many microcontroller programming courses, and he has developed skills in many aspects of electronics.

Brandon Kincheloe

Brandon joined the United States Army as a 14E PATRIOT Missel Defense Operator/Maintainer in 2002. He honorably ETS'ed from the army in 2005. He attained the National Defense Service Medal, Global War on Terrorism Service Medal, and Army Service Ribbon. In 2005, Brandon began his life as a civilian planning on using my GI-Bill for education but decided to enter the general workforce. He worked a number of different jobs such as landscaping and factory work from 2005 to 2009. In 2009, Brandon started working at Wal-Mart. In 2012 he obtained his American Board of Opticianry (ABO) certification. In 2014 he became a student at Fort Scott Community College (FSCC). While there, Brandon was a Phi Theta Kappa (PTK) member, and STEM Club President. He graduated from FSCC in spring of 2018, Summa Cum Laude with my Associates of Science & Pre-Engineering degree. In Fall of 2018, Brandon continued his education at Pittsburg State University (PSU). He majors in Electrical Engineering Technologies with an emphasis in Embedded Systems. While at PSU, Brandon joined the National Society of Leadership and Success (NSLS) and was a member of Pi Kappa Phi (PKP). He attained a number of achievements at PSU such as first place in the 2021 research colloquium undergraduate division featuring my research in deep learning. Brandon specializes in rapid prototyping design techniques such as 3D design and printing and circuit board PCB development cycles. Brandon graduated May 16th of 2022 with many honors including Summa Cum Laude and EET Outstanding Senior 2022.

Donna Lenharth (Lecturer)

Donna Lenharth is a Systems Analyst at Names and Numbers in Pittsburg, KS and a Lecturer in the Engineering Technology department at Pittsburg State University. She has a Bachelor of Science in Computer Science from the University of Missouri-Rolla where she graduated Magna Cum Laude. She has worked as a software developer at IBM, BMC Software, and Textron. While at Textron, she received the Chairman's Award for Innovation for her work on Future Combat System. Donna was recently awarded a NASA scholarship through the Kansas Space Grant subaward and is pursuing a Master of Science in Mathematics.

Kailash Chandra (Professor)

Kailash Chandra is a Professor at Pittsburg State University in Kansas where he has taught Computer Science and Engineering Technology courses for forty years. His research interests include machine learning, programming languages, and data structures. He previously taught at Quincy University in Illinois. He received his education at Meerut University, Florida Tech, and SUNY Buffalo.

Introducing Deep Learning on Edge Devices Using A Line Follower Robot

Abstract

An educators' job is to prepare students to work with the latest technology. In the last decade, there has been an increase in the use of deep learning in many applications including self-driving cars and virtual assistants. This paper discusses the teaching of the fundamentals of deep learning in an undergraduate microcontroller which used the TI – RSLK MAX robot which is based on an ARM-based MSP432 microcontroller. In the course, a new concept of deep learning was introduced each week and included hands-on activities. The deep learning portion of the course culminated with a project in which the students were assigned to use deep learning in a line follower robot. In this paper, the students are co-authors and they provided data about their robots. This data was used to assess the effectiveness of the deep learning curriculum.

Introduction

Machine learning, a subfield of artificial intelligence, is concerned with creating algorithms that allow a computer to learn from data. This is useful in many applications such as face recognition processing, computer vision, and natural language processing. Deep learning is a subset of machine learning and can involve large amounts of data. Deep learning involves the use of a neural network that mimics the biological brain. The neural network is trained by examples as opposed to being explicitly programmed. Deep learning makes it possible for computers to learn very complex models that were previously difficult to learn. This allows engineers to build many applications and solve problems that were hard or impossible to solve in the past.

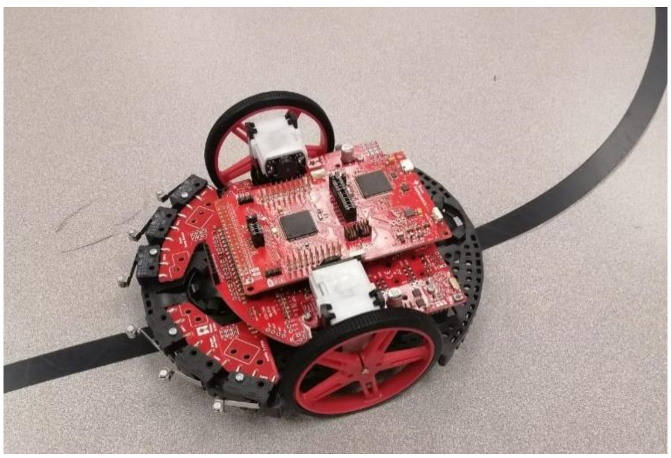

Fig.1: TI – RSLK MAX

This paper describes a hands-on approach to teaching students deep learning and involves creating line follower robots that use deep learning. This curriculum was used for undergraduate and graduate microcontroller courses in the Electronics Engineering Technology and Master of Engineering Technology programs at Pittsburg State University. The curriculum material was designed for students with a basic knowledge of programming. During the laboratory activities, students learned to program deep learning using Google Colaboaratory which is a programming tool that enables the building and training of deep learning projects [6]. Users can use different hardware accelerators, such as Graphics Processing Unit (GPU) or Tensor Processing Unit (TPU), so students could get high-speed results without expensive computers [1]. Students also learned to use Keras, which is a popular Python library that is used with neural networks [5].

Methods

Neural networks learn by example, similar to human learning. For deep learning, there are typically three or more layers in a neural network: an input layer, one or more hidden layers, and an output layer as shown in Fig. 2. Each layer consists of neurons and the layers can have one or many neurons. Moreover, the size and number of hidden layers will depend on the application of the neural network. For example, an application may need to distinguish between clothing like a t-shirt, jacket, and stockings. If after training the model and the results were insufficient, one solution is to add more neurons or hidden layers.

A layer of a neural network is defined as being *dense* if each neuron in the layer is connected to every neuron in the previous layer. The output layer, hidden layer 1, and hidden layer 2 in Fig. 2 are dense layers.

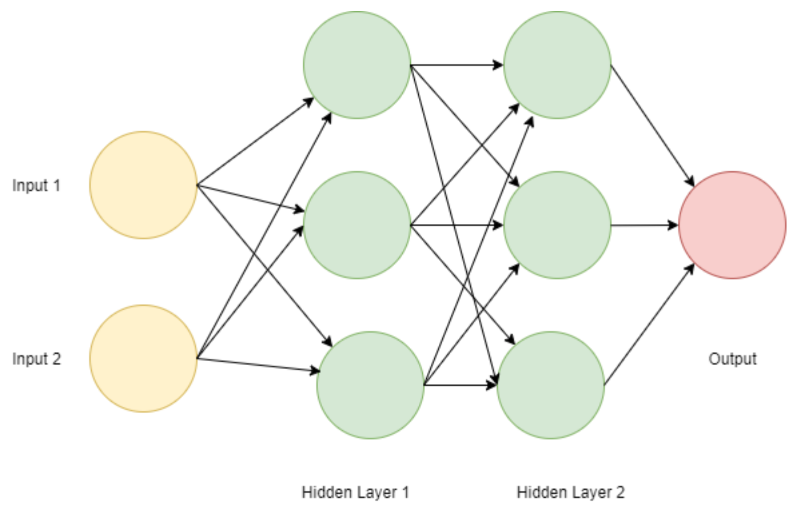

Fig. 2: A simple deep learning network with two layers

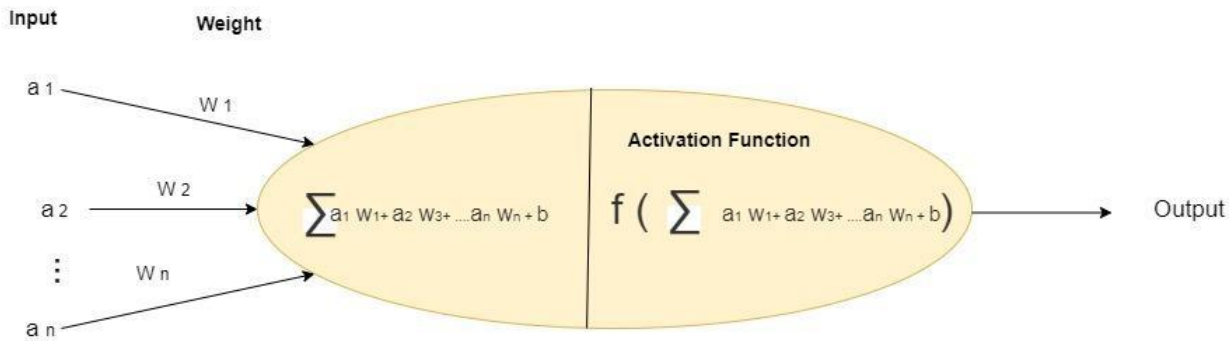

Fig. 3: Internal view of neuron

In a neuron, each input will have a weight associated with it as shown in Fig. 3. The weights are multiplied by the inputs, the products are summed, and a bias is added. The result is typically used as an input to an activation function. The output of the neuron will be the output of the activation function.

The activation function will depend on the application of the neural network. In a linear regression, the neural network is trained to output numbers in response to inputs that are also numbers. The Rectified Linear Unit (ReLU) and sigmoid activation functions can be used for linear regression and are shown in Figs. 4 [3] and 5 [4].

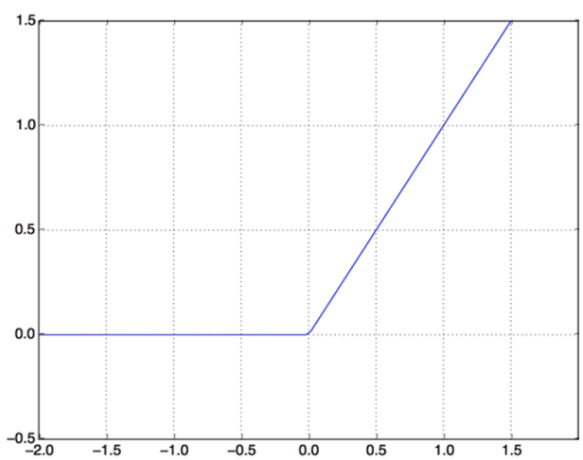

Fig. 4: ReLU activation function [3]

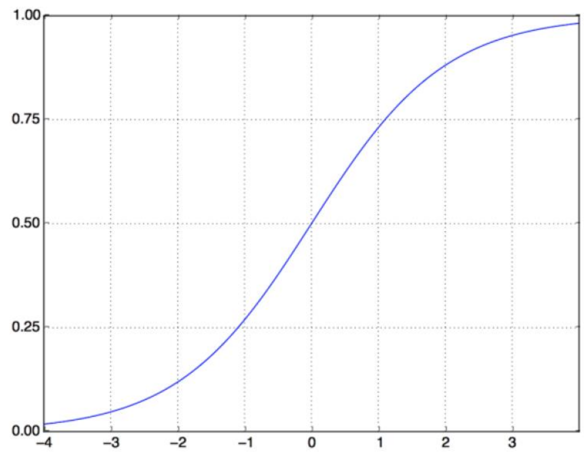

Fig. 5: Sigmoid activation function [4]

The *Rectified Linear Unit* (ReLU) [5] activation function returns the input value if the input is greater than zero. Otherwise, the output will be zero. The output can be written:

$$\text{ReLU} = \text{Max}\,(x,0) \qquad (1)$$

where x is the input [7].

The *sigmoid* activation function returns values between 0 and 1 and the output can be calculated from:

$$sigmoid(x) = \frac{1}{(1+e^{-x})} \qquad (2)$$

where x is the input [7].

In a classification application of a neural network, the network classifies the inputs into several categories. An example is using pixel values from images as inputs and classifying the pictures as a cat or dog. One-hot can be used in classification. For one-hot, only one of the categories is 1 for each input while the remainder are zero. For the cat vs. dog example, the one-hot outputs could be either (0, 1) for a cat and (1, 0) for a dog.

The softmax activation function can be used in classification and converts the floating-point outputs of a layer into probabilities between 0 and 1 where the summation of the probabilities of the outputs is equal to1 (3). The output of neuron *i* will be calculated by [2]:

$$\sigma_i = \frac{e^{y_i}}{\sum_{j=0}^{N} e^{y_j}} \qquad (3)$$

The neural network is trained using training data [5]. The *loss function* indicates how well a neural network is trained. The mean squared error function is typically used for linear regression applications. The *mean squared error* function calculates the square root of the difference between the predicted and true values [8].

$$\text{Loss} = \text{Mean}(\ (y_{true} - y_{pred})^2) \tag{4}$$

where y_{true} are the desired outputs of the neural network and y_{pred} are the current outputs of the neural network. For classifier applications, the is cross-entropy function is commonly used as the cost function [10].

The algorithm that adjusts the weights of the neural network to minimize the loss function during training is called the optimizer. One successful optimizer is the *Adam* optimizer. The *learning rate* indicates how large a change should be made in the weights. An *epoch* is one adjustment of the weights using the training data.

Line follower robot

The line follower robot project was used in the Spring 2022 semester in an undergraduate class on microcontrollers. Each week, a new concept of deep learning was introduced along with a new lab activity. The weekly schedule for the course is shown in Table 1.

Table 1: Undergraduate curriculum with lab activity

WEEK	CLASS CONTENT & LAB ACTIVITY
Week 1	Introduction to Deep learning and Google Colaboratory
Week 2	Training neural networks and linear regression
Week 3	Sigmoid and ReLU activation functions
Week 4	Classification
Week 5	Convolutional neural networks
Week 6	Recurrent neural networks
Weeks 7-10	Line follower robot with deep learning
Weeks 11-16	Individual projects

Texas Instrument’s TI – RSLK MAX shown in Fig. 1 was used for the line follower robot with deep learning [9]. The TI – RSLK MAX is based on an ARM-based MSP432 microcontroller and has a line sensor array with eight infrared LED/phototransistor pairs for the training data, the inputs were the digital signals from the line sensor array and the outputs were chosen to control the left and right motors such that the robot will follow a line. Training the neural networks was done on cloud servers using Keras and Google Colaboratory. After the neural network was trained, it was deployed onto the microcontroller with a C program.

The line follower robot was used as a lab activity as shown in Table 1. Students were provided access to Google Colaboratory, a TI – RSLK MAX, and Keil IDE. The line follower robot project was assigned at the beginning of March 2022. To assess the students’ projects, the rubric in Table 2 was applied.

Table 2: Robot performance assessment

Category	Robot continuously stayed on line	Robot occasionally left line	Robot occasionally followed line	Robot under development	Weighted Average
Weight	3	2	1	0	
Number of robots	4	1	0	1	2.33
Robot designation	A, B, D, E	F	-	C	-

Assessment

The students in the undergraduate microcontroller course were asked if they would like to be co-authors for this paper. The student co-authors provided data on their line follower robots that they developed. At the time of the final draft of this paper, the course was still in progress. Thus, some of the line following robots were still under development.

A line for testing the robots was made from a black electrical tape and consisted of straight and curved segments as shown in Figs. 6 and 7. Six-line follower robots were used to collect the assessment data and four robots were correctly following the line as shown in table 2 while one of the robots occasionally left the line. Details on the neural networks used in the robots is summarized in Tables 3-7.

Table. 3: Neural network for Robot A

Layer	Number of neurons	Activation function	Comments	Optimizer	Learning rate	Epochs	Loss function	Loss
input	1		Input is 8-bit integer.	Adam	0.0005	First 500	Mean squared error	-
0	1	None	Dense.					
1	3	Sigmoid	Dense.		0.0001	Last 10,000		
2 (output)	1	None	Dense. Linear regression: output is floating point with range 1 to 3 Output is rounded up and 1 = turn right 2 = forward 3 = turn left					

Table. 4: Neural network for Robot B

Layer	Number of neurons	Activation function	Comments	Optimizer	Learning rate	Epochs	Loss function	Loss
input	1		Input is 8-bit integer.	Adam	0.001	40,000 - 50,000	Mean squared error	-
0	1	ReLU	Dense.					
1	2	Sigmoid	Dense.					
2 (output)	1	None	Dense. Linear regression: output is floating point with range -3 to 3 and used as continuous value with -3 = hard left 0 = straight 3 = hard right					

Table. 5: Neural network for Robot C

Layer	Number of neurons	Activation function	Comments	Optimizer	Learning rate	Epochs	Loss function	Loss
input	1		Input is 8-bit integer.	Adam	0.1	500	Mean squared error	-
0	10	ReLU	Dense.					
1	6	ReLU	Dense.					
2 (output)	1	None	Dense. Linear regression; output is floating point with range 1-3. Output is rounded up and 1 = turn left 2 = forward 3 = turn right					

Table. 6: Neural network for Robot D

Layer	Number of neurons	Activation function	Comments	Optimizer	Learning rate	Epochs	Loss function	Loss
input	1		Input is 8-bit integer.	Adam	0.1	500	Mean squared error	0.0125
0	4	ReLU	Dense.					
1	4	ReLU	Dense.					
2 (output)	1	None	Dense. Linear regression: output is floating point with range 1-3 used to calculate PWM					

Table. 7: Neural network for Robot E

Layer	Number of neurons	Activation function	Comments	Optimizer	Learning rate	Epochs	Loss functio n	Loss
input	8		8 single-bit inputs	Adam	0.1	500	Mean squared error	1.046×10^{-5}
0	3	ReLU	Dense.					
1	4	ReLU	Dense.					
2 (output)	3	SoftMax	Dense. Classification: output is one-hot and categories were left, straight, and right.					

Table. 8: Neural network for Robot F

Layer	Number of neurons	Activation function	Comments	Optimizer	Learning rate	Epochs	Loss functio n	Loss
input	8		8 single-bit inputs	Adam	0.1	100	Mean squared error	2.384×10^{-8}
0	3	ReLU	Dense.					
1	3	ReLU	Dense.					
2 (output)	3	None	Dense. Classification: used highest output to decide category which was left, straight, or right.					

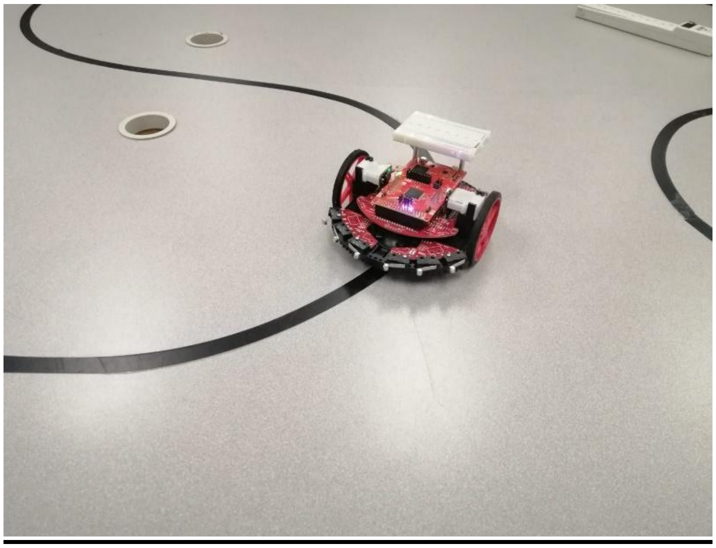

Fig.6: Working Robot

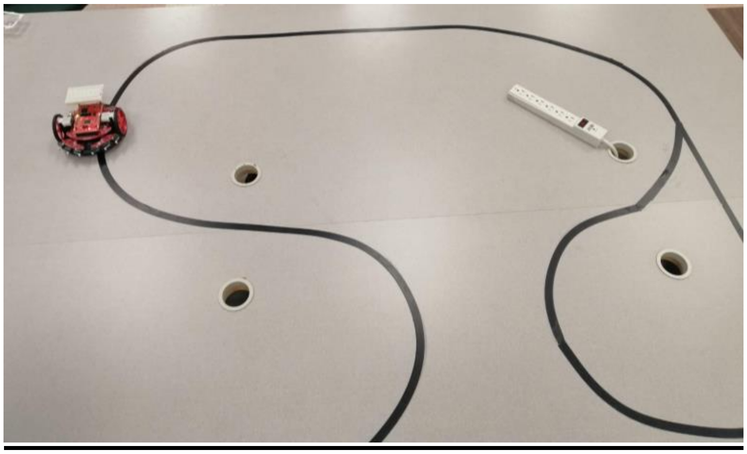

Fig.7: Working Robot

Conclusion

There has been a significant rise in deep learning applications over the years. To engage students to learn and build these applications, this paper introduced a hands-on approach to teaching deep learning. Students were able to learn the fundamentals of deep learning and built a line follower robot. At the time of the final draft of this paper, there were three out of six robots that followed a line successfully and the others were under development. Assessment results showed that half of the students successfully built a deep learning application. Even though students used deep learning for a relatively simple line follower robot, they gained valuable such as how to program deep learning applications and implement them into a microcontroller. Many applications can be built based upon the techniques that the students have learned throughout the course.

References

[1] Ashley, K. (2020). *Applied Machine Learning for Health and fitness a practical guide to machine learning with deep vision, sensors, Iot, and Vr*. Apress.

[2] *A Quick Introduction to Artificial Neural Networks (Part 2) – Kris Bolton*. (n.d.). Krisbolton.com. https://krisbolton.com/a-quick-introduction-to-artificial-neural-networks-part-2

[3] Brownlee, J. (2020, August 20). *A gentle introduction to the rectified linear unit (ReLU)*. Machine Learning Mastery. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/#:~:text=The%20rectified%20linear%20activation%20function,otherwise%2C%20it%20will%20output%20zero.

[4] Chollet, F. (n.d.). *Deep Learning.* Manning. 2020

[5] *Deep Learning with TensorFlow | Udacity Free Courses*. (n.d.). Www.udacity.com. https://www.udacity.com/course/intro-to-tensorflow-for-deep-learning--ud187

[6] Google Colab. (n.d.). https://colab.research.google.com/

[7] Team, K. (n.d.). *Keras documentation: Layer activation functions*. Keras.io. https://keras.io/api/layers/activations/

[8] Team, K. (n.d.). *Keras documentation: Regression losses*. Keras.io. https://keras.io/api/losses/regression_losses/#mean_squared_error-function

[9] *TI robotics system learning kit max (TI-RSLK max)*. TI Training. (2020, December 28). https://training.ti.com/ti-robotics-system-learning-kit-max

[10] Wikimedia Foundation. (2022, May 4). *Cross entropy*. https://en.wikipedia.org/wiki/Cross_entropy

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration under Grant No. 80NSSC20M0109. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration nor of Wichita State University.