Article **Journal Innovative Design**
December, 2022 Vol.6 No.15 19-25

MAC-based Artificial Neural network for voice command recognition

Red Neuronal Artificial basada en MAC para reconocimiento de comandos de voz

RODRÍGUEZ-PONCE, Rafael†*

Universidad Politécnica de Guanajuato, Ingeniería en Robótica

ID 1st Author: *Rafael, Rodríguez-Ponce* / **ORC ID:** 0000-0001-5006-5580, **CVU CONACYT ID**: 209261

DOI: 10.35429/JID.2022.15.6.19.25 Received July 23, 2022; Accepted October 30, 2022

Abstract

Artificial neural networks are one of the most popular families of machine learning algorithms of this decade. Although they exist since the middle of the last century, until recent years with the improvement of technology, they are being widely used in fields such as character, image, and voice recognition. There is a large number of works implementing neural networks for speech recognition; however, the approach has usually been for operation on a personal computer, which is not suitable for mobile applications. This article presents a neural network for voice command recognition, implemented in a compact FPGA card with low computational resources. In addition, it uses a multiplication and accumulation unit, called MAC, with which it achieves a smaller size and higher speed. This paper will be of interest to students or researchers working on machine learning mobile applications.

Automatic speech recognition, Mel-frequency cepstral coefficients (MFCC), Field programmable gate array (FPGA)

Resumen

Las redes neuronales artificiales son en una de las familias de algoritmos de machine learning más populares de esta década. Aunque existen desde mediados del siglo pasado, hasta recientes años con la mejora de la tecnología, están siendo ampliamente utilizados en campos tales como reconocimiento de caracteres, imágenes y voz. Existe un gran número de trabajos de implementación de redes neuronales para reconocimiento de voz, sin embargo el enfoque normalmente ha sido para funcionamiento en una computadora personal, lo cual no es adecuado para aplicaciones móviles. En este artículo se presenta una red neuronal para reconocimiento de comandos de voz, implementada en una tarjeta FPGA compacta y de bajos recursos computacionales. Además, utiliza una unidad de multiplicación y acumulación, denominada MAC, con la cual logra un menor tamaño y mayor velocidad. Este documento será de interés a estudiantes o investigadores interesados en aplicaciones móviles de machine learning.

Reconocimiento automático de voz, Coeficientes cepstrales de frecuencias de Mel, Matriz reconfigurable de compuertas digitales

Citation: RODRÍGUEZ-PONCE, Rafael. MAC-based Artificial Neural network for voice command recognition. Journal Innovative Design. 2022, 6-15: 18-25

*Correspondence to the Author (e-mail: rrodriguez@upgto.edu.mx)
† Researcher contributing as first author.

Introduction

Nowadays, we live in a time when digital electronics are an integral part of our lives. They are found in computers and communication equipment, household devices, entertainment equipment, and power tools, among others; however, in some instances it would be desirable for the device or equipment to be able to carry out, in some way, a learning process without the need for user intervention, in other words, to contain some type of artificial intelligence.

An artificial neuron is a mathematical-computational model with various inputs and outputs, which tries to emulate the functioning and learning of a biological neuron (Hudson and Cohen, 2012). Thanks to the latest advances in computing technology, they have begun to be used mainly in digital applications such as character recognition (Liu, *et al*., 2020), image classification (Dong, *et al*., 2022), text generation (Zhang, *et al*., 2019), language translation (Nguyen, *et al*., 2019) autonomous driving (Chen, *et al.*, 2021), and many others; however, an application that has attracted great interest in recent years is voice-command recognition (Nassif, *et al.*, 2019).

For the implementation of a neural network, it is common to use a personal computer; however, for mobile applications, a more compact, lightweight electronic system with low power consumption is necessary. For these cases, a microcontroller or an FPGA is the most recommended option, although they are more limited in computational resources, compared to a computer. (Ma, Cao and Sao, 2018).

The main operation of a neural network can be summarized as a weighted sum of all the values of the network inputs. Thus, its implementation requires a large number of addition and multiplication operations. If the network is very large all the arithmetic resources of the system can be exhausted in a small device. However, using a more robust device implies a greater consumption of time and power. (Zu y Sutton, 2003).

Although multiple implementations of neural networks have already been made for voice command recognition, a personal computer with robust graphics processing units (GPU) and using high-level programming languages are normally used. Even though they have provided excellent results, they are not suitable for being incorporated into mobile technology (Lyashenko, *et al*., 2021) due to all the computational resources and electrical power they require.

In this work, it was decided to implement an artificial neural network for voice-command recognition on an FPGA since, despite having low computational resources; it allows the design of user-customized digital architectures. In this project, the system is based on the use of a single Multiply and Accumulate (MAC) unit for each neuron, allowing a low number of arithmetic elements used.

A MAC unit is a digital architecture that allows the weighted sum required by the neural network, to be executed cyclically. So no matter how many inputs the neuron has, it always uses a single MAC unit (Nedja, *et al.*, 2009). Furthermore, the FPGA already has these high-speed MAC units embedded in its digital architecture, so there is no need to use logic gates for its implementation.

There are different methods for extracting relevant information from the voice signal (Cabral, Fikai and Tamura, 2019); however, one that has shown very good results is the Mel Frequency Cepstral Coefficients (MFCC). These coefficients are obtained through a filter bank whose operation is similar to that of the human ear (Jo, Yoo and Park, 2015).

In this paper, a network of six single-layer neurons was implemented in an FPGA, for the identification of six voice commands. First, the voice signals are processed using MFCC to extract the most relevant information. Then, these coefficients are fed to a network of six neurons with eight inputs each. The network was trained with 4,000 voice commands from men and women, adults and children, achieving a general accuracy greater than 95%.

ISSN-2523-6830

This work can be very interesting and useful for students, teachers, or researchers interested in the implementation of neural networks for voice command identification in mobile devices.

Artificial Neuron

An artificial neuron also called a perceptron, is a computational function, which has several data inputs and a single output. The neuron uses this input data to perform a weighted sum of them. This weighting is given by a weight that is assigned to each of the input connections, and in this way, changes the intensity with which each input affects the behavior of the neuron. It is also assigned a bias value, which is represented as an internal connection of the neuron. After the sum of the input/weight products and the bias, an activation function such as a sigmoid is applied, which is a non-linear function and, in a certain way, distorts the output value so that gradually, large values converge to 1 and small values converge to 0. The structure of an artificial neuron is shown in Figure 1.

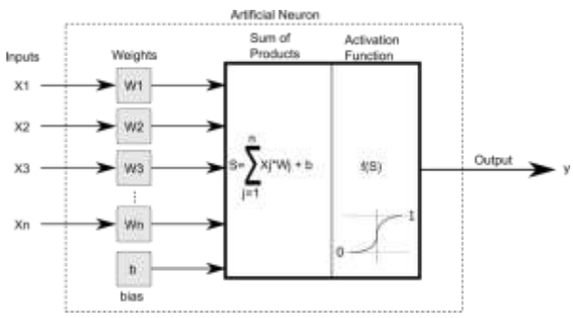

Figure 1 Structure of an artificial neuron
Source: Self-Made.

The importance of the activation function occurs mainly when several neurons are chained together to form a neural network that achieves a deeper level of learning. Thus, the non-linearity of each neuron allows a more complex function. Otherwise, it would behave such as a simple linear regression function (Pomerat, Segev and Datta, 2019).

Mel Frequency Cepstral Coefficients

A widely used technique for speech recognition is Mel Frequency Cepstral Coefficients or MFCC. This method is used to extract the most relevant information from the audio signal and feed it to the artificial neuron.

For the application of this technique to voice commands, Librosa was used, which is a computational package in the Python programming language designed for the analysis of music and audio signals. (McFee, *et al.*, 2015).

The information extraction procedure is described below and shown in Figure 2. First, the Discrete Fourier Transform (DFT) is applied to obtain the spectral content of the audio signal, passing from the time domain to the frequency domain. It is then passed through a filter bank on the Mel scale, which extracts the most relevant information in certain frequency ranges, which is similar to the functioning of the human ear. This scale is linear below 1 kHz and logarithmic above this threshold (Jo, Yoo and Park, 2015). Finally, the logarithm of these energies and the discrete cosine transform (DCT) are obtained to again obtain information in the time domain, similar to the inverse Fourier transform, except that the coefficients obtained with the DCT they are always real numerical values.

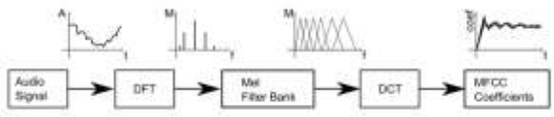

Figure 2 Block diagram for the MFCC technique
Source: Self-Made.

MAC Unit Architecture

A MAC unit is a digital structure that is used to perform a sum of products of any length, using a single addition and multiplication operation cyclically. The benefit of using these structures is that it allows savings in arithmetic resources in an FPGA. (Mohindroo, Paliwal and Suneja, 2020).

The digital structure of a MAC unit is shown in Figure 3 and described as follows. Firstly, there is the input of two data values which are multiplied arithmetically. The result of the product is now added to the contents of an accumulator, which is zero in the very first cycle, and this result in turn is stored in the accumulator.

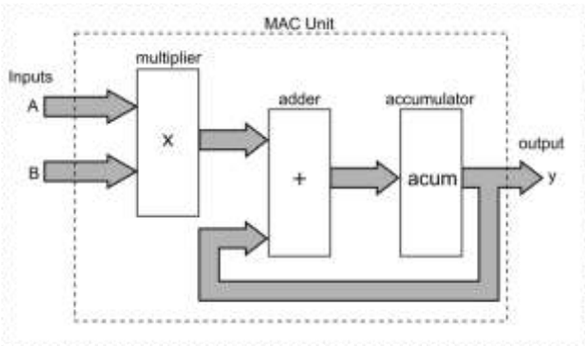

Figure 3 MAC unit structure
Source: Self-Made.

In the next cycle, the next two data inputs are multiplied and added to the current contents of the accumulator, which are then stored in the accumulator for the next cycle. This process is repeated indefinitely, depending on the number of data values to be multiplied. Once the multiplication and addition process is completed, the final result can be obtained at the output of the MAC unit. It is worth mentioning that, for the control of input data and process repetitions, a Finite State Machine (FSM) is used, which is a sequential digital structure commonly applied for process control in digital systems.

In Figure 4(a) the complete diagram of the MAC unit is shown, and Figure 4(b) shows the control logic that follows the FSM of this unit, which is described as follows. To start the multiplication and accumulation process, the required number of N cycles is indicated to the FSM and a start signal is given. Each input data is multiplied and added with the content of the accumulator register, and the FSM sends a signal so that this result is stored in the register again. A digital counter keeps track of the required cycles and, at the same time, outputs this data, which will be used to synchronize the input of the MFCC coefficients to the neuron. Once the process is finished, the FSM sends a signal to an output register so that it allows the final result to be output, and a completion signal is also sent to the next block, to indicate that the result can be retrieved.

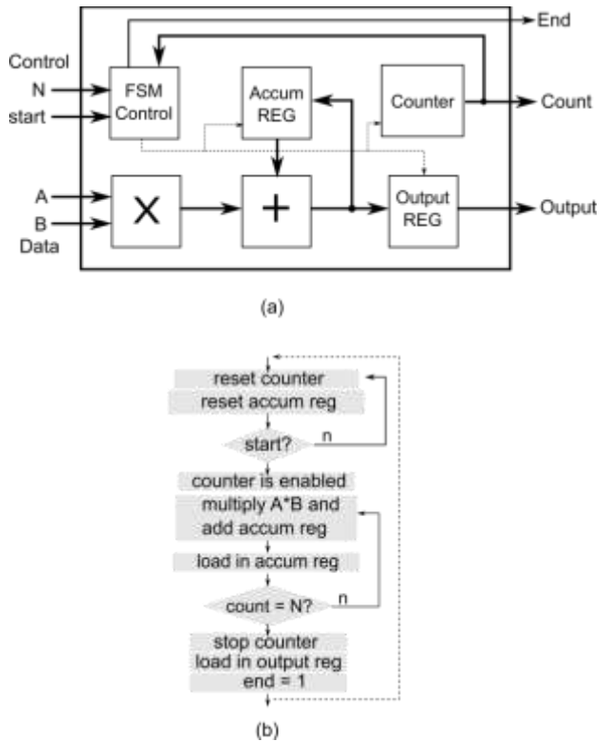

Figure 4 (a) Complete structure of the MAC unit and (b) the corresponding FSM logic control
Source: Self-Made.

The main advantage of the MAC structure is that a sum of products of any size can be achieved with a single multiplier and adder. Despite being cyclical, it can be carried out at a very high speed when implemented in an FPGA. (Mohindroo, Paliwal and Suneja, 2020).

System architecture

The description of the complete neural network functionality has been divided into two parts for ease of understanding. Firstly, the digital architecture required for data input to the neuron is shown when the neuron training is disabled, and later, when training is enabled.

Neural network architecture with training disabled

When the neuron training is disabled, only the products of the coefficients and the weights are carried out, the sigmoid activation function is applied and the result is sent to the output (Figure 5). This process is controlled by an FSM which is in charge of synchronizing the input of weights and coefficients to the MAC unit so that, one by one, they are multiplied and accumulated.

The initial weights used are random values stored in a Look-Up Table (LUT), which are only updated when the neuron training is enabled. Once the sum of products operation is finished, the bias value is added, which initially is also a random value and is updated in training mode.

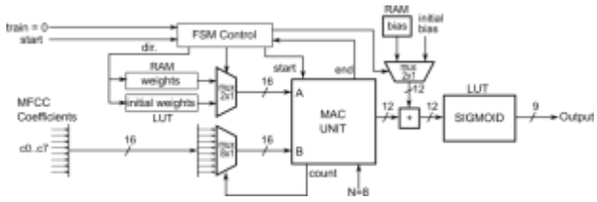

Figure 5 Digital structure for the artificial neuron with training disabled
Source: Self-Made

Finally, the data is passed to a LUT for the sigmoid function, which has 12 input bits and 9 output bits.

Neural network architecture with training enabled

When the neuron training is enabled, in addition to the weighted sum and the application of the sigmoid function, the error between the current output data and the desired output is calculated. This error value is used to carry out the correction of the weights and the bias so that the error tends to be close to zero.

First, an adjustment factor is calculated using gradient descent. This correction value is added to the weights and bias and stored back in memory to be used in the next training cycle. Again, the FSM is responsible for controlling the entire process of error calculation, gradient descent, and weight/bias correction. (Figure 6).

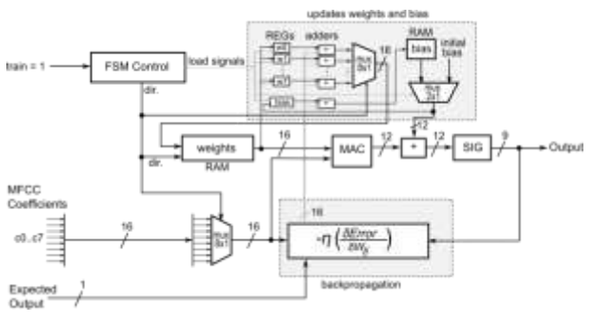

Figure 6 Digital structure for the artificial neuron with training and learning mode enabled
Source: Self-Made

It is worth mentioning that the input and output width of the neuron is completely generic, so they can be easily changed by simply adjusting the parameter at the time of structure generation, without the need for code modification.

Complete Neural Network

The architecture shown above in Figure 6 represents the neuron for a single voice command. In the case of this paper, six voice commands were selected, for which the network is made up of six identical neurons (Figure 7), which are executed and trained simultaneously. At the output of the network, there is a magnitude comparator, which outputs a logical 1 for the signal with the highest magnitude, and a logical 0 for all others.

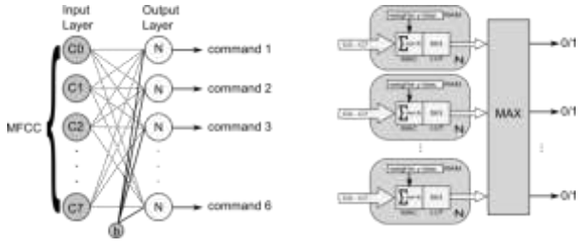

Figure 7 Digital structure for the complete neural network
Source: Self-Made

Testing and Results

In the first instance, 4000 voice signals of six different commands in Spanish were collected, these being: stop, forward, up, down, left, and right. The intention of using these six commands is for a future project of voice control for an unmanned aircraft or drone.

Voice commands were collected from men and women of different ages, in a closed environment without noise. All voice commands were set to a 1-second duration, being digitally recorded on a cell phone or personal computer. The information preparation process is shown in Figure 8 and described below.

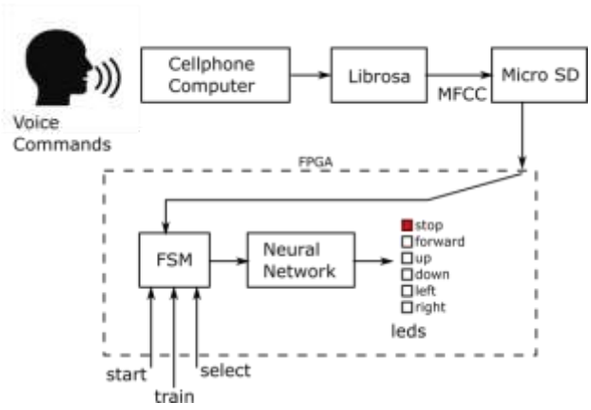

Figure 8 Voice command preparation and processing process
Source: Self-Made

First, the voice signals were processed in Python using the Librosa libraries for the extraction of the MFCC coefficients and stored in a microSD digital memory card. It is worth mentioning that, in this application, it is possible to select the number of MFCC coefficients to extract. However, the greater the number of coefficients, the greater the number of inputs required in the neuron and, therefore, the greater the number of operations to be carried out in the FPGA. After several tests, it was found that eight MFCC coefficients were more than enough to properly distinguish each of the voice commands.

Subsequently, the microSD memory is inserted into the FPGA containing the artificial neuron (Figure 7), and the training process of the neural network is started using 2000 of the voice commands stored in memory. These commands are fed one by one and are processed at high speed so that 5 seconds were enough to train the entire network.

For the network validation, the remaining 2000 voice commands were used, which are fed to the neuron every second, to be verified in the FPGA card's LEDs. The success rate for voice commands is shown in Table 1.

Commands	Voiced	Successful	Percentage
Stop	330	318	96.3%
Forward	330	315	95.4%
Up	330	321	97.2%
Down	330	325	98.4%
Left	350	334	95.4%
Right	330	323	97.8%

Table 1 Success rate for each voice command
Source: Self-Made

The FPGA board used is an Intel-Altera DE1-SoC, with a Cyclone V and a 150 MHz clock frequency.

Conclusion

In this work, a neural network for the identification of six voice commands was presented. This network was implemented in an FPGA using a single MAC unit for each neuron. With this architecture, the identification of the commands coming from adults and children of both sexes was achieved, with a success greater than 95%.

This work will be of great interest to students, teachers, or researchers in search of a simple and efficient alternative for the implementation of neural networks.

References

Cabral F.S., Fukai H. and Tamura S. (2019). Feature Extraction Methods Proposed for Speech Recognition Are Effective on Road Condition Monitoring Using Smartphone Inertial Sensors, *Sensors, 19*(1), 1-20. DOI: 10.3390/s19163481.

Chen L., Lin S., Lu X., Cao D., Wu H., Guo C., Liu C. and Wang F.Y. (2021). Deep Neural Network Based Vehicle and Pedestrian Detection for Autonomous Driving: A Survey, *IEEE Trans. Int. Transp. Sys., 22*(6), 3234-3246. DOI: 10.1109/TITS.2020.2993926.

Dong Y., Liu Q., Du B. and Zhang L. (2022). Weighted Feature Fusion of Convolutional Neural Network and Graph Attention Network for Hyperspectral Image Classification, *IEEE Trans. Image Proc., 31*(1), 1559-1572. DOI: 10.1109/TIP.2022.3144017.

Hudson D. and Cohen M. (2012). *Neural Networks and Artificial Intelligence for Biomedical Engineering*. Wiley-IEEE Press, ISBN: 978-0-470-54535-5. DOI: 10.1109/9780470545355.

Jo J., Yoo H. and Park I.C. (2015). Energy-Efficient Floating-Point MFCC Extraction Architecture for Speech Recognition Systems, *IEEE Trans. VLSI Syst., 24*(2), 754-758. DOI: 10.1109/TVLSI.2015.2413454.

Liu X., Hu B., Chen Q., Wu X. and You J. (2020). Stroke Sequence-Dependent Deep Convolutional Neural Network for Online Handwritten Chinese Character Recognition, *IEEE Trans. Neural Netw. Learn Syst., 31*(11), 4637-4648. DOI: 10.1109/TNNLS.2019.2956965.

Lyashenko V., Laariedh F., Sotnik S. and Ahmad M.A. (2021). Recognition of voice commands based on Neural Network, *TEM Journal*, *10*(2), 583-591, DOI: 10.18421/TEM102-13.

Ma Y., Cao Y. and Sao J.S. (2018). Optimizing the Convolution Operation to Accelerate Deep Neural Networks on FPGA, *IEEE Trans. VLSI Syst.*, *26*(7), 1354-1367. DOI: 10.1109/TVLSI.2018.2815603.

McFee B., Raffel C., Liang D., Ellis DPW., McVicar M., Battenberg E. and Nieto O. (2015). *Librosa: audio and music analysis in python*, In Proceedings of the 14th python in science conference, 18-25. DOI: 10.5281/zenodo.6097378.

Mohindroo B., Paliwal A. and Suneja K. (2020). *FPGA-based Faster Implementation of MAC Unit in Residual Number System, INCET* International Conference on Emerging Technology, Balgaum, India. DOI: 10.1109/INCET49848.2020.9154105

Nassif A.B., Shahin I., Attili I., Azzeh M. and Shaalan K. (2019). Speech Recognition Using Deep Neural Networks: A Systematic Review, *IEEE Access*, *7*(1), 19143-19165. DOI: 10.1109/ACCESS.2019.2896880.

Nedja N., Da Silva R.M., Mourelle L.M. and Da Silva M.V.C. (2009). Dynamic MAC-based architecture of artificial neural networks suitable for hardware implementation on FPGAs, *Neurocomputing*, *72*(1), 2171-2179. DOI: 10.1016/j.neucom.2008.06.027.

Nguyen Q.P., Vo A.D., Shin J.C., Tran P. and Ock C.Y. (2019). Korean-Vietnamese Neural Machine Translation System with Korean Morphological Analysis and Word Sense Disambiguation, *IEEE Access*, *7*(1), 32602 – 32616. DOI: 10.1109/ACCESS.2019.2902270.

Pomerat J., Segev A. and Datta R. (2019). On Neural Network Activation Functions and Optimizers in Relation to Polynomial Regression, *IEEE Int. Conference on Big Data*, Los Angeles, Calif. USA. DOI: 10.1109/BigData47090.2019.9005674.

Zhang R., Wang Z., Yin K. and Huang Z. (2019). Emotional Text Generation Based on Cross-Domain Sentiment Transfer, *IEEE Access*, *7*(1), 100081 - 100089. DOI: 10.1109/ACCESS.2019.2931036.

Zu J. and Sutton P. (2003). *FPGA Implementations of Neural Networks - A Survey of a Decade of Progress*, Ed. Springer-Verlag, Berlin, 1062-1066. DOI: 10.1007/978-3-540-45234-8_120.