Performance Evaluation of Neural Networks for Animal Behaviors Classification: Horse Gaits Case Study

E. Cerezuela-Escudero, A. Rios-Navarro, Juan P. Dominguez-Morales, R. Tapiador-Morales, D. Gutierrez-Galan, C. Martín-Cañal and A. Linares-Barranco

Abstract The study and monitoring of wildlife has always been a subject of great interest. Studying the behavior of wildlife animals is a very complex task due to the difficulties to track them and classify their behaviors through the collected sensory information. Novel technology allows designing low cost systems that facilitate these tasks. There are currently some commercial solutions to this problem; however, it is not possible to obtain a highly accurate classification due to the lack of gathered information. In this work, we propose an animal behavior recognition, classification and monitoring system based on a smart collar device provided with inertial sensors and a feed-forward neural network or Multi-Layer Perceptron (MLP) to classify the possible animal behavior based on the collected sensory information. Experimental results over horse gaits case study show that the recognition system achieves an accuracy of up to 95.6%.

Keywords Multi-Layer Perceptron · Feed-forward neural network · Pattern recognition · Inertial sensors · Sensor fusion

1 Introduction

Behavior monitoring of wildlife animals is a hard technological task [1] due to several factors that need to be solved. (1) The development a lightweight and long batteries life (thus, low power consumption) devices to attach to the animal; (2) The design and implementation of a wireless network to collect the information

E. Cerezuela-Escudero · A. Rios-Navarro · J.P. Dominguez-Morales(✉) ·
R. Tapiador-Morales D. Gutierrez-Galan · C. Martín-Cañal · A. Linares-Barranco
Robotic and Technology of Computers Lab, ETS Informática - Universidad de Sevilla,
Av. Reina Mercedes s/n, 41012 Sevilla, Spain
e-mail: {ecerezuela,arios,jpdominguez,ricardo,dgutierrez,cmartin,alinares}@atc.us.es

from those devices; and finally, (3) download from the animals devices all captured information. Some commercial devices can track animals using global position systems (GPS) and obtain some of their vital signs through sensors. The information given consists only of raw and unprocessed data that require high bandwidth communications or high capacity memory cards and long life batteries, which usually are very heavy. Furthermore, these solutions are not able to recognize animal patterns from the obtained data.

Algorithms that look for particular behavioral patterns based on the input data usually conduct this kind of recognition or classification. Some of these algorithms are Neural Networks (NN), Support Vector Machines (SVM) or even complex statistical methods, which can detect specific behaviors such as sleeping, running, copulating, etc. Generally, the computational costs of these algorithms are highly enough to require specific platforms capable of parallelizing computations for this classification.

In this paper, a particular NN architecture is implemented for a behavioral classification of wildlife animals restricted to horse gaits, which are the ways a horse can move. The NN is designed and trained using a software tool and then all of its parameters are obtained and used on an embedded NN version implemented to run on a low-power microcontroller. There are some hardware platforms like SpiNNaker [2] that allows the easy development of spiking neural networks; however, the size and power consumption of this board are extremely high for this kind of task and targets.

MINERVA is a research project whose main aim is to study and classify wildlife behavior inside Doñana National Park [3]. To achieve this goal, a hierarchical wireless sensor network capable of transmitting and storing this information has to be set and tested inside this park. Moreover, in order to increment the value of the final product, an embedded system with energy harvesting techniques that will be able to digest sensor fusion data from inertial sensors [4], combined with other sensors (temperature, heart rhythm, GPS) has been developed in order to classify animal behavior in real time. This project has the additional aim of developing an infrastructure for collecting this information and make it accessible through the internet. The pattern recognition of the sensed data is performed in real time by the microcontroller using a low-power implementation of a NN that classifies three different horse gaits [12] (motionless, walking and trotting). This information is transmitted using a wireless multisensory network distributed on collars placed on some animals. This multisensory network reads data from the sensors and send them to a network of motes, which acts as a router and retransmits these packets to a base station. This base station receives the information through the network and uploads it to a remote server database. Researchers can access this data using a web-based user interface and track the animal activity at any time without the necessity of being in Doñana National Park.

The NN implementations presented can classify three different horse gaits. MATLAB (a mathematical software), with a toolbox for NN, has been used in order to design the architecture. The NN has been trained with data collected in Doñana with a prototype of the collar configured to capture and store raw data with a parameterized period of time. The NN training has been performed in the same software using 70% of the collected data. The remaining 30% has been used for the NN testing.

The paper is structured as follows: section 2 presents the collar device. Then, section 3 describes different fusion filters applied to the sensor data. Section 4 presents the NN architecture to classify three different behaviors. Section 5 describes the testing scenario and the results obtained. Finally, section 6 presents the conclusions.

2 Collars: Information Collection by Multiple Sensors

The aim of this collar is to gather information from the animal on which it is placed by using several sensors. Then, it will classify its animal behavior using this data as an input for a feed-forward NN implemented on the collar microcontroller. All detected patterns are locally stored in the collar memory. Finally, the collar will send the recognized gaits to a base station through XBee communications that will upload it to a database stored in a data server on the internet. The collar is provided with an inertial measurement unit (IMU), which consists of a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer. This unit is used in addition to a GPS, which gives the position and time with respect to satellites. The IMU used in this work is MinIMU-9V2 [5], whose sensors have a resolution of 12 bits. The feed-forward NN architecture and training process is described in section IV.

The collar prototype, see Fig. 1, has an XBee module (XBee PRO S2B [6]) that can transmit data through a wireless network. XBee is the brand name for Digi International for a family of form factor compatible radio modules. XBee modules are integrated solutions based on ZigBee, which is an open global standard of the IEEE 802.15.4 MAC/PHY [7]. This device family allows to implement a mesh network of motes (or routers) where collars (or device) send information, and other elements (coordinators) of the network redirect these packets to a web server. The main objective is to transmit sensed information to the nearest router of the network, so that it can reach the coordinator and upload this information to the database. In such a case the signal cannot reach a valid point to transmit, i.e. the animal is out of the network coverage, the collar carries an SD card where the information is stored; so the animal behavioral information can be accessed later or offline, so it avoids data losing.

The periodical measurements of each sensor are carried out using a low power microcontroller (STM32L152 [8]) with a real-time operating system (RTOS) which is powered using a four AAA battery pack (1.5V, 1155 mAh each). Due to the fact that capturing an animal is very expensive, the process of obtaining data from each collar when it runs out of battery is organized as a task, allowing the microcontroller to switch to sleep mode if there are no router in the network coverage capable of receiving this collar's information. This increases batteries life. Moreover, the collar does not spend the whole time transmitting the information in a continuous manner. A periodic time is established for reducing radio transmissions and thus, reducing power consumption.

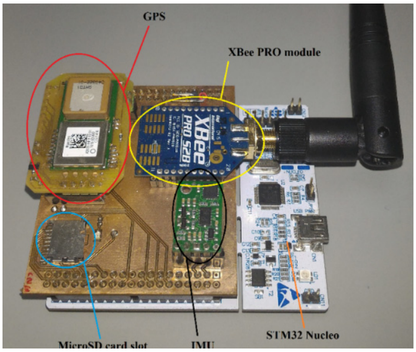

Fig. 1 Collar device prototype

3 Sensors Data Processing

Information from the sensors is enough to develop a NN able to recognize patterns from animals. Additionally, it is possible to increase the accuracy if more data are used during training. Sensor fusion obtains new combined information from sensors that can simplify and increase the accuracy of the NN architecture. This new information consists of three variables called pitch, roll and yaw, which are the angles of rotation of each axis of a three-dimensional space (pitch represents y-axis rotation, roll represents x-axis rotation and yaw the variation from z-axis). The purpose of this fusion is to generate new data for the NN and then compare results using data from sensors and from fusion algorithms.

An accelerometer is a sensor that measures the gravity acceleration at 3D each axis (x,y,z). It is possible to obtain pitch and roll calculation by applying mathematical formulas based on the combination of data collected by the accelerometer; however, if the accelerometer is not precise enough, then, small variations produce high signal-to-noise ratio (SNR) values. On the other hand, it is difficult to calculate the yaw value without combining the magnetometer tilt compensation and the gyroscope variation.

This paper uses two fusion algorithms: (1) FreeIMU [9] and (2) Kalman Filter [10]. FreeIMU, Fig. 2, is based on a quaternion representation. A quaternion is a complex number that represents the object orientation by four fields. The three first fields represent the orientation in each axis, while the last field represents the rotation of the object. The algorithm fuses the accelerometer and the magnetometer using the gradient descent algorithm when the fusion is finished. The gyroscope data are added to compensate the possible drift. Finally, in order to get pitch, roll and yaw, the values of the quaternion are combined. The second algorithm is the Kalman Filter, Fig. 2, which is a complex sensor fusion algorithm commonly used in control systems [10]. The principal advantage of this algorithm is that, by using initials estimators and dynamic parameters, the algorithm auto adjusts the output in time.

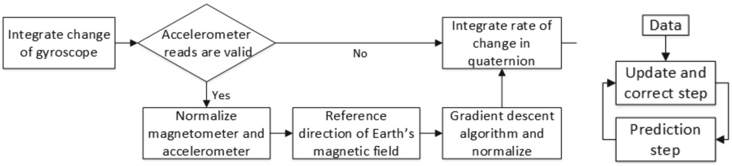

Fig. 2 FreeIMU filter (left) and Kalman filter (right)

The prediction process generates an estimation of the next value and calculates the estimated error covariance. The second step updates the current value from the sensors and the error covariance using the different parameters mentioned before and the estimations calculated in the previous step. When each value is updated, it is sent to the prediction step. This feedback continues updating the error estimation and value predictions, so the algorithm converges over iterations. By using this algorithm it is possible to filter noise from each sensor generating pitch, roll and yaw. The main difference between these algorithms is the computational cost: Kalman filter is more accurate but requires a higher computational effort; on the other hand, FreeIMU does not have any feedback so it does not change filter parameters over time, adjusting the output to the data from the sensors. The idea of implementing both algorithms is to compare the results of the classification obtained in the NN and, with this output, decide which one is more adequate for an animal behavior classification.

4 Neural Network

4.1 Neural Network Architecture

This section presents the architecture of the used Multi-Layer Perceptron (MLP) NN. MLP is the most commonly used with the backpropagation algorithm: the multilayer feed-forward network. An elementary neuron with R inputs is shown in Fig. 3. Each input p is weighted with an appropriate w. The sum of the weighted inputs and the *bias* forms the input to the transfer function f [11]. Neurons can use any differentiable transfer function f to generate their output.

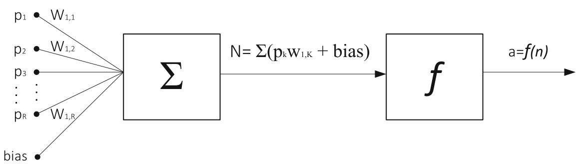

Fig. 3 Multi-layer Perceptron neuron

The topological structure of the MLP NN used consists of a two-layer feed-forward network, with a sigmoid transfer function in the hidden layer and softmax transfer function in the output layer. During this research, the optimal number of hidden units was found by running different performance tests, where a new MLP was created, trained and tested using a varying number of neurons in the hidden layer. The best performance was obtained using 10 neurons in the hidden layer. The number of neurons in the output layer is equal to the number of behaviors to be classified. In this manuscript, three different patterns are presented: motionless, walking and trotting. The results are presented for different input data: the raw data acquired from the IMU sensors (x, y and z for each sensor) and the output from the fusion algorithms presented before (roll, pitch and yaw). Therefore, the number of inputs in the NN directly depends on which data are used at the performance test. The network was trained using a scaled conjugate gradient backpropagation algorithm [13].

4.2 Neural Network Input and Target Data

The NN input data was the IMU sensors data collected from the collar while a horse performed three kinds of behaviors: motionless, walking and trotting. The sensors described in section 2 gathered data every 0.03 seconds, during a specific time period of 1 second (this is configurable), obtaining 2000 samples for each behavior (sampling frequency of 33 Hz).

The first performance test used the instantaneous raw sensor data and, consequently, the NN had *nine* inputs (x, y and z for each 3-axis sensor of the IMU). To evaluate the utility of each sensor on the recognition phase, the NN was tested with different combinations of sensors and using different numbers of neurons in the hidden layer.

On the other hand, for the second and third performance tests, Kalman and FreeIMU algorithms were used. In these two cases, the number of inputs in the NN were three (pitch, roll and yaw).

Finally, as forth performance test, to evaluate the utility of the frequency spectrum information for animal pattern recognition, the Discrete Fourier Transform of nine sensor data was calculated using a Fast Fourier Transform (FFT). The data used for training and testing the NN were the single-sided spectrum. Thus, the number of NN inputs were also nine and corresponds, for each frequency step, to the nine signals (3-axis of 3 sensors of the IMU).

5 Experimental Results and Analysis

This section presents the experimental results of the classification system with the MLP neural network varying the number of neurons in the hidden layer and the input data between raw and different filtered sensor data.

5.1 Classification System Results Using Sensor Raw Data (Unprocessed)

The classifier system was trained and evaluated using 30,000 samples (10,000 samples of each kind of behavior) obtained from the accelerometer, gyroscope and magnetometer. The samples were randomly divided into three sets: 70% for training, 15% for validation and 15% for testing. Table 1 shows the classification results for the raw sensor data testing set, using different numbers of neurons in the hidden layer.

Table 1 Proportions of true class accuracies using MLP and unprocessed sensor data

Neurons in Hidden Layer	Classes	Sensors used				
		Accelerom.	Gyroscope	Magnet.	Acc. Gyr. Mag.	Acc. Gyr.
10	Trotting	82.4%	78.2%	74.2%	84.0%	82.9%
	Motionless	82.8%	66.0%	43.6%	85.3%	80.7%
	Walking	67.6%	21.0%	53.7%	73.6%	71.4%
	Average	**77.6%**	**55.1%**	**57.2%**	**81.0%**	**78.3%**
20	Trotting	82.9%	80.4%	72.5%	86.0%	83.1%
	Motionless	83.0%	71.0%	46.9%	88.3%	86.8%
	Walking	69.8%	17.5%	57.7%	75.1%	70.9%
	Average	**78.6%**	**56.3%**	**59.0%**	**83.1%**	**80.3%**
30	Trotting	82.0%	77.2%	73.2%	88.4%	85.7%
	Motionless	84.9%	76.3%	49.3%	90.7	85.8%
	Walking	68.6%	22.2%	59.4%	78.0%	74.1%
	Average	**78.5%**	**58.6%**	**60.6%**	**85.7%**	**81.9%**

These results show that the accelerometer is the sensor with better information about the horse movement, while the gyroscope and magnetometer improve the pattern definition. The classifier system has an accuracy of 85.7% with 30 neurons in the hidden layer.

5.2 Classification System Results Using Kalman and FreeIMU Filters

The classifier system was trained and evaluated using 30,000 samples of pitch, roll and yaw. These samples were obtained when applying Kalman and FreeIMU filters to accelerometer, gyroscope and magnetometer raw data in real-time when the microcontroller collar captured this data. The samples were randomly divided into three sets: 70% for training, 15% for validation and 15% for testing. Table 2 shows the classification results for the testing set, using different numbers of neurons in the hidden layer and the applied filter.

Table 2 Proportions of true class accuracies using MLP with filtered sensor data

Neurons in Hidden Layer	Classes	Applied filter	
		Kalman	FreeIMU
10	Trotting	100%	70.5%
	Motionless	93%	51.0%
	Walking	93.9%	57.0%
	Average	**95.6%**	**59.5%**
20	Trotting	99.9%	71.4%
	Motionless	93.6%	57.9%
	Walking	93.7%	66.2%
	Average	**95.7%**	**65.2%**

From Table 2, the recognition performance of our classification system using Kalman filter is 95.6% regardless of the number of neurons in the hidden layer. Therefore, the best performance was obtained with at least 10 neurons in the hidden layer.

5.3 *Classification System Results Using FFT Filtered Sensor Data*

In order to calculate the FFT of the IMU sensor data, we divided the samples in sets of 256 and 512 samples. The classification system was trained and evaluated using the FFT data calculated for both cases. The samples were randomly divided into the same groups as in 5.2. Table 3 shows the classification results for the testing set, using different numbers of neurons in the hidden layer.

Table 3 Proportions of true class accuracies using MLP and FFT

Neurons in Hidden Layer	Classes	Number of samples for FFT	
		256	**512**
20	**Trotting**	77.8%	79.9%
	Motionless	59.7%	60.5%
	Walking	47.9%	45.3%
	Average	**61.8%**	**61.9%**
30	**Trotting**	80.5	80.2%
	Motionless	62.7	61.4%
	Walking	51.4	48.9%
	Average	**64.8%**	**63.5%**

These experimental results show that the best accuracy of the recognition system using accelerometer, gyroscope and magnetometer sensors and a MLP neural network was obtained by processing the sensor data with the Kalman fusion algorithm. Two approaches can be considered: to use a Cortex-M4 family microcontroller with a FPU using Kalman filter data as input, sacrificing battery life; or to use a low-power microcontroller without a FPU to save on battery consumption (using raw data as input).

6 Conclusion

In this work, we propose a system to recognize animal behaviors based on artificial-intelligent devices with inertial sensors, based on a NN implementation to classify the possible horse gaits from the collected sensors information. To evaluate the classification system accuracy, four performance tests with different sensor data processing have been performed. The sensors fusion algorithms used were Kalman and FreeIMU. In each test, a MLP NN was created, trained and tested using a varying number of neurons in the hidden layer. The best average accuracy value is 95.6% and it is obtained using 10 neurons in the hidden layer and the Kalman Filter. FreeIMU fusion algorithm and FFT do not bring any improvement to the accuracy of the recognition system. In the case of raw sensor data, the MLP NN needs 30 neurons in the hidden layer to attain 85.7% success. Future work will increase the number of behaviors and animals, and study historic aggrupation of data over time for the classification. The use of FANN open-source library for NN implementation on microcontrollers is under evaluation.

Acknowledgement This work was supported by the excellence project from Andalusia Council MINERVA (P12-TIC-1300).

References

1. Dominguez-Morales, M., et al.: Technical viability study for behavioral monitoring of wildlife animals in Doñana. In: Proc. Int. Conf. Data Commun. Netw. Int. Conf. Opt. Commun. Syst., pp. 98–101 (2011)
2. Painkras, E., et al.: SpiNNaker: A 1-W 18-Core System-on-Chip for Massively-Parallel Neural Network Simulation. IEEE J. Solid-State Circuits **48**, 1943–1953 (2013). doi:10.1109/JSSC.2013.2259038
3. Doñana National Park: http://whc.unesco.org/en/list/685
4. Tapiador-Morales, et al.: System based on inertial sensors for behavioral monitoring of wildlife. Int. Conf. Comput. Inf. Telecommun. Syst. (2015)
5. MinIMU-9 v2. https://www.pololu.com/product/1268
6. XBEE Pro SB2. http://ftp1.digi.com/support/documentation/90000976_W.pdf
7. IEEE 802.15.4 - IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) (2011)
8. STM32L152RET6. http://www.st.com/web/catalog/mmc/PF259539
9. Madgwick, S.O.H.: An efficient orientation filter for inertial and inertial/magnetic sensor arrays 2010. doi:10.1109/ICORR.2011.5975346
10. Chen, S.Y.: Kalman filter for robot vision: A survey. IEEE Trans. Ind. Electron., 4409–4420 (2012)
11. Haykin, S.: Neural Network: A Comprehensive Foundation, 2nd edn. Prentice Hall (1998)
12. Harris, S.E.: Horse Gaits. Balance and Movement. Howell Book House, New York (1993). ISBN 0-87605-955-8
13. Moller, M.F.: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks **6**, 525–533 (1993)