Proceedings of the World Congress on Engineering 2008 Vol I
WCE 2008, July 2 - 4, 2008, London, U.K.

Neural Network Application for High Speed Impacts Classification

Israel Gonzalez-Carrasco, Angel Garcia-Crespo, Belen Ruiz-Mezcua, Jose Luis Lopez-Cuadrado *†

Abstract—**This paper presents the research carried out in order to obtain the most efficient neural network design, that allows to approach the steel armours response under high speed projectile impact. One of the main problems related to neural networks is the high quantity of data needed for their training and testing, as well as the complexity of their numeric simulation. In the domain of ballistic impact the large number of parameters involved in the problem hampers the simulation of a large number of impact cases. Trying to solve this issue, the developed research established as aims on the one hand to minimise the number of data used for training, and on the other hand to analyse the influence of the different training variables or input parameters on the learning ability of the created network. The results obtained highlight the small number of data needed to obtain acceptable results, as well as the clear influence of some variables on the generalization ability of the network for this domain.**

Keywords: ***Neural network, steel armour, ballistic impact, terminal ballistics, numerical simulation***

1 Introduction

There are a wide number of systems which, during their service life, can suffer the impact of objects moving at high speed (over 500 m/s). This phenomenon is named impact ballistic and the most characteristic examples are found in the military field. However over the last decades, this kind of problems has become of interest in civil applications. In them, the structural elements are required to absorb the projectile energy so that it does not damage critical parts of the global system.

Due to this, there are new possibilities in similar fields, among which passive safety vehicle stands out. In this field, it is especially relevant the design of structures whose mission is to absorb energy in crashes of Crashworthiness type (200 m/s $\leq$ speed $\leq$ 500 m/s), as well as those that can appear in road or railway accidents, helicopters' emergency landings, etc. Therefore, what is being sought is to design structures capable of absorbing energy to avoid or lessen the damages to the passengers of the concerned vehicles.

The construction of structures subjected to impact was traditionally carried out empirically, relying on real impact tests each using the given projectile/target. The mathematical complexity of solving the equations that rule the impact phenomenon, and the relative ignorance of the mechanical behaviour of the materials at high strain rates, discouraged any simulation of the problem.

The need for design tools to simulate this process triggered the development in recent years of a large number of models of different types; all of them belong to two families: those of analytical modelling and those of numerical simulation. Thus, the use of expensive experimental tests has been postponed to the final stage of the design. All the previous stages can be covered by the use of this kind of simulation tools.

Taking into account the difficulties of these methods, poor precision and high computational cost, a neural network for the classification of the result of impacts on steel armours was designed. Furthermore, the numerical simulation method was used to obtain a set of input patterns to probe the capacity of the model development. In the problem tackled with, the available data for the network designed include, the geometrical parameters of the solids involved - radius and length of the projectile, thickness of the steel armour - and the impact velocity, while the response of the system is the prediction about the plate perforation.

2 Ballistic Impact

The ballistic terminology is defined as the scientific study of everything related to the movement of the projectile. Within this science, the Terminal Ballistic or Effect Ballistic discipline stands out. It is in charge of studying the results produced in the body or object that suffers the impact of the bullet or projectile. Moreover it analyses the way the projectile behaves when it reaches its target, how the projectile ends up, how the transference of kinetic energy is carried out and what effects it has on

*Manuscript received April 21, 2008. This research was done with the financial support of the Comunidad Autonoma de Madrid under Project GR/MAT/0507/2004.

†Department of Computer Science, Universidad Carlos III, Av. Universidad 30 - 28911. Leganes (Madrid). Spain. Tel.: +34 91 624 9117. Fax.: +34 91 624 91 29. I. Gonzalez-Carrasco email is: ig-carras@inf.uc3m.es. Others authors email are: acrespo@ia.uc3m.es, bruiz@inf.uc3m.es, jllopez@inf.uc3m.es

the objective, etc.

The conditions in which this projectile impacts on the objective depend most of the times on different parameters, namely the materials, both of the projectile and the target, the speed, the impact angle, the thickness of the target, the diameter of the projectile and finally the shape and weight. The combination of these factors leads to the existence of a vast variety of possible behaviours once the contact between them is produced.

The model developed was simplified to reduce the computational complexity of the numeric simulations needed to recreate impact tests of this type. For this purpose the obliquity of the projectile was not taking into account. Therefore, a right angle between the axis of the projectile and the surface of the plate where it impacts against, has been considered.

From a purely physique point of view, the impact of a projectile on a protection can originate two different results. The process of the projectile's entrance inside the protection during an impact is called penetration.

A perforation is produced when the projectile manages to go through the protection completely. On the other hand, an arrest takes place when the projectile goes inside the protection but does not manage to go through it.

Finally if the angle of attack in the impact comes into play, the projectile can rebound, what is known as ricochet.

The investigation carried out in this study tries to analyse the behaviour of metallic projectiles when the impact against metallic plates, both recreated with the same SAE 1006 steel material. This material has been chosen because it is isotope, its parameters are known and it allows carrying out further tests in the laboratory at a low cost.

3 Use of ANN in Impact Situations

Artificial Neural Networks (ANNs) are statistical models of real world systems which are built by tunning a set of parameters. These parameters, known as weights, describe a model which forms a mapping from a set of given values, the inputs, to an associated set of values, the outputs.

The mathematical model that the neural network builds is actually made up of a set of simple functions linked together by the weights. The weights describe the effect each simple function (known as unit or neuron) will have on the overall model [1].

Within the field of research of this article, these neural network types have been applied successfully within the limits of the fracture mechanic [2] to estimate material breakage parameters such as concrete [3], and in no-destructive tests to detect breaks in more fragile materials [4]. In all these applications, the input data needed for the training has been obtained by means of experimentation and numeric simulation.

Nevertheless there are not enough studies dedicated to the application of Neuronal Networks to problems of ballistic impacts. At the present time the investigations made have focused on low speed impacts or situations where it is necessary an energy absorption, Crashworthiness [5, 6, 7].

Therefore, and due to the mentioned experience on the mechanical problems and on the good results shown on the previous researches [8], a MultiLayer Perceptron (MLP) neural network with backpropagation algorithm has been defined. The most important attribute of this kind of ANN is that it can learn a mapping of any complexity or implement decision surfaces separating pattern classes.

An MLP has a set of inputs units whose function it is to take input values from the outside, a set of outputs units which report the final answer, and a set of processing hidden units which link the inputs to the outputs. The function of the hidden neurons is to extract useful features from the input data which are, in turn, used to predict the values of the output units.

MLPs are layered neural network, that means they are based on several layers of adaptive weights and neurons. There are three different layers: on the one hand the input layer, on the other hand at least one hidden layer and finally the output layer. Between the units or neurons that compose the network there are a set of connections that link one to each other. Each unit transmits signals to the ones connected with its output. Associated with each unit there is a output signal or transference signal, that transform current state of the unit into an output signal.

The feedforward connections between the units of the layers in a MLP represent the adaptive weights that permit the mapping between the input variables and the output variables [9].

Different authors have showed that MLP networks with as few as a single hidden layer are universal approximators. In other words ANN are capable to approximate with accurate arbitrary regions, if they have enough hidden units [10].

3.1 Impact Scenario Parameters

Within the limits this article deals with, there are different variables or parameters that characterize the behaviour of the projectile when it impacts on an steel armour. Therefore, there are various parameters to shape the input data to the network, being the latter ones the ones that define the number of neurons of the input layer.

The available variables are: kinetic energy (K), velocity (V), radius (R), length (L), mass (M) and quotient L/R, being all of them related to the projectile, and on the other hand the thickness of the objective (H).

However, using all the available variables is not always necessary to carry out the training. In some cases, an information overload or the existing connections between the variables can saturate the prediction model that adjusts the output of the network, therefore complicating the learning and reducing the generalization rates.

In this domain, K is correlated with V and M. On the other hand, the ratio L/R and M are correlated with R and L because density is constant. So for the neural network performance it is better to remove K and ratio L/R from the list of available variables.

Moreover, in order to support the latter assertion within the limits of the problem dealt with, a series of studies were designed to measure the influence that each variable has, separately and in connection with the remainder, on the learning and the network generalization ability.

3.2 Network Structure

For the network design, a three level MLP architecture was selected. According to Lippman's studies, this type of structure allows to solve most of the problems, being able to form any complex random limit of decision [11].

The number of inputs in a MLP is determined by the number of available input parameters in the problem dealing with. The number of outputs is the same as the number of classes needed to solve the problem.

The number of hidden layers and the number of neurons of these layers have to be chosen by the designer. There is not a method or rule that determines the optimum number to solve a problem given. In some cases these parameters are determined by test and error. However, there are current techniques for obtaining automatic estimations [12], although this research follows the steps described by Tarassenko [13].

The neural network was created with the commercial software Neurosolutions for Excel v4 and its architecture is shown in Figure 1.

- 5 neurons in the input layer linked with the five input variables (R,L,H,M,V). The chosen transference function is the identity.
- 4 neurons in the hidden layer with hyperbolic tangent transference function.
- 2 neurons in the output layer: associated to the output variable plate perforation (both outputs are complementary as it allows to improve the learning process). The chosen transference function in this case is the hyperbolic tangent.

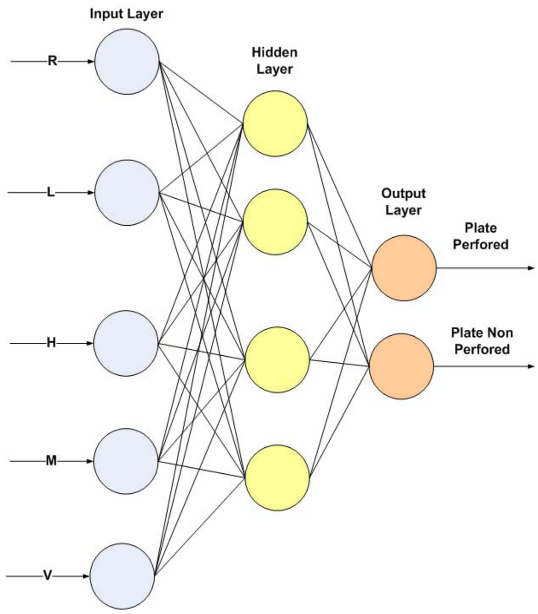

Figure 1: Structure of the neural network designed

3.3 Input Data

It is important to make an estimation of the necessary information to train the network in an appropriate way. For this purpose, it is essential to have a number of data large enough to ensure that. While the network is trained, it is essential to recognise and answer to the wide set of conditions that may appear during the testing process. If the number of data is too small, the complete range of connections that the neural network should learn will not be covered.

There are different approximations to calculate the optimum number of input data [9]. The designed system has followed the guidelines set by Tarassenko to calculate the number of examples needed [13].

Several impact problems were generated from which some of them were used for the training and testing process and the other for the validation of the results obtained by the network after learning. The modelling and simulation were done with the commercial software ABAQUS/Explicit v6.4.1 [14], a finite element explicit code widely used in highly non-linear dynamic simulation.

The stopping criterion used was fixed in the previous researches made by the authors [8]. A specific number of epochs to train the network, 10000, has been chosen,

ISBN:978-988-98671-9-5

so the neural network does not use validation data in the training process because the stopping criterion is set by the epochs and not by the cross validation error. So the ANN only uses a train subset for the training process and a test set to validate the accurate of the network.

In this case, the set of training patterns is repeatedly presented in random order to the network and the weight update equations are applied after the presentation of each pattern. When all the patterns have been presented once to the network, one epoch of training is made.

After the training process, the unseen patterns of the test set are presented to the network. This demonstrates the generalization performance of the trained network.

4 Solution Proposed

The initial aim established for this study was to find the network design capable of approximating most rightly whether a projectile perforates or does not perforate in a steel armour. This was achieved by the authors in a previous research, finding a prediction model with a very high ability of success, which needed few training data to obtain good generalization rates [8].

In the light of the results obtained, the neural networks present themselves as an alternative to be borne in mind regarding the impact problem. The network results reliable to predict the projectile arrest with the advantage, opposing to the use of simulation tools, that it boasts a computational cost very inferior once the training has been carried out.

However, these conclusions lead to another question. Could there be a more efficient ANN model? That is to say a model that would need less input variables and input data to be able to approximate the output function correctly.

The research carried out to solve this question established two fundamental goals, on the one hand to minimize the number of training data without affecting the generalization ability of the model; and on the other hand, to analyse the influence of the different input variables on the network learning ability.

These input variables form the independent variables of the function that allows to approximate the perforation function. The result of this answer function is a dichotomic variable that depends on the input variables, and admits "Yes" or "No" values.

Hence it is possible to infer that the easier the function that describes the network behaviour is, the less costly will generate the set of training data through numeric simulation be.

The type of heuristic selection of input variables and training set size selected is well documented in the early literature of Neural Networks within other domains, and the results obtained certifies the suitability of this election [15].

4.1 Randomness Elimination

The results obtained in the former researches [8], despite being positive, present the uncertainty if they depend on a possible randomness of the data intended for training and testing. In other words, if the assignment of the available data influence on the predictions of the network.

To carry out a research with conclusive results, it was established a number of trials by far superior to the previous work. The study carried out is broken out in a series of experiments, in which the number of patterns intended for training and test varies in each series. This way, the minimum number of data needed to ensure an acceptable percentage of correct answers, can be found.

It has been accomplished 100 different trials in each experiment, exchanging in each of them and in a random way the data destined to train and test the network. Moreover, in two different trials of a same experiment the same training and test data will never coincide, therefore increasing the reliability on the results obtained. Thanks to this, the possible random data that are provided to the network, which could lead to obtain not very reliable results, are eliminated.

The catalogue of experiments to be done in the study is the following: the first one includes 100 trials with 50 training data and 10 testing data. From this point onwards, in each of them the number of data intended to training is reduced to 5. The last experiment consists of 100 trials with 15 training data and 10 testing data.

The result obtained in each one of these 100 trails is processed and the average error porcentage is showed for each experiment. In this way, the results obtained in each of the 8 experiments can be compared. Thanks to this, it can be determined how many training data are necessary to achieve good generalization rates; and besides, it ensures that the possible random data of the input data do not influence on these results.

4.2 Determination of the Influence of the Independent Variables

The second part of this study is focused on the analysis of the influence of each variable within the training process and, therefore, of the mathematical model that regulates its output.

So the main question is to find the explanatory variables, in other words, to include in the model just those variables that can influence on the answer, rejecting those ones that do not provide information. It is also necessary, look for the possible interactions between the independent variables that affect the answer variable. For example, it this work it will analyse the results obtained by including the thickness variable and the velocity variable separately, as well as when they are both included.

5 Results Obtained

The data shown in Table 1 and Table 2 are the average error percentages that the network makes for each experiment. The different experiments are grouped in columns, where it is indicated the parameters or variables used, e.g the column "All-H" points out that the experiment uses all the variables except the parameter H.

One result example in Table 1, the value 3,2, is the average error of the network with all the parameters for 100 batteries of test with 50 data to learn (train and test) and 10 to testing.

Taking into account the objective sought, to find the network architecture that makes the smallest error, when analysing the data it could be established that the more input variables the network has, the better it learns.

However, it can occur that the network saturation, that is to say to introduce all the available magnitudes, is not the best option. In this problem, this network architecture has the lowest error probability only for 50 and 45 batteries. In addition to this, there are two designs that have lower average probability of error, the ones without mass and without length. Specifically the design that omits the mass as a network learning variable is the one that has the smallest average error.

On the other hand, for the average of the error obtained, it can be observed that the velocity is a very relevant magnitude for the network learning. Its omission leads the error probability to increase 342%, or what is the same, 4,4 times in relation to the network design considered as the most efficient (network architecture without mass).

Finally as expected, the network with the worst results is the one with less information, that is, with less input variables. However, it is advisable to highlight that the network without the thickness and mass variables has quite acceptable results (12,91%) (see Table 2).

Taking the best configuration for the most architectures, 50 train and 10 testing, the Figure 2 depicts the average error percentage made for each network architecture in which some variable is removed with regard to the network that holds all of them.

Table 1: Error percentage obtained by varying the input variables and the number of training data

Train. Data	All	All -H	All -L	All -M	All -R
50	3,2	7,7	3,3	4	5,6
45	5,3	12	6,5	5,9	6,2
40	5,3	15,3	4	5,7	6,5
35	5,6	10	6,9	5,5	4,8
30	11	14,1	6,6	7,3	9,7
25	11,2	11,2	9,5	8,8	9,4
20	13,4	14,2	14,6	11,8	14,2
15	13,9	11,9	13,6	10,6	13,8
Average	8,61	12,05	8,13	7,45	8,78

Table 2: Error percentage obtained by varying the input variables and the number of training data

Train. Data	All -V	All -H&M	All -H&V	All -H,M&V
50	26,3	10,2	30,1	37,9
45	34,1	10	33	45,6
40	30,1	10,6	36,5	38,4
35	27,2	11,3	36,3	39,4
30	32,9	14,5	42,6	43,1
25	36,6	16,5	39	46,6
20	35,7	17,3	39,1	44,1
15	40,7	12,9	42,8	45,2
Average	32,95	12,91	37,43	42,54

6 Conclusions

In the light of the results and taking into account that perceptron is one of the simplest topologies, this work shows clearly the possibilities of the neural networks in the prediction of the material behaviour at high deformation speeds.

On the other hand, the architecture chosen presents a high reliability when predicting the result of a projectile impact. Moreover, its computational cost once the training has started, is smaller than the one of the simulations carried out with tools of finite elements.

It is crucial to highlight the small number of training data needed for the network to learn with a relative small error in its predictions. With only 40 numeric simulations of finite elements, the network obtains an error below 6% in both designs with the best results (with all the variables and without the mass variable). In spite of the small number of input data, the network does not present overlearning problems.

The experiments developed help to better understand the ballistic impact, analysing the influence of the parameters that appear in the penetration problem. The results obtained verify that the variable with the most influence is velocity. Furthermore, any network architecture where

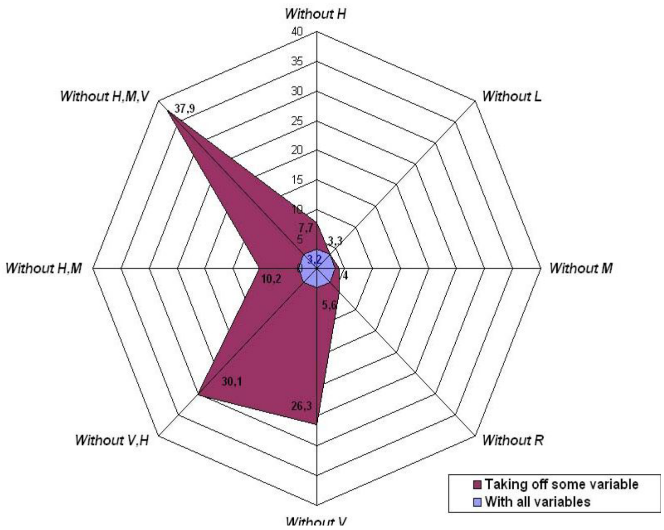

Figure 2: Error made in each architecture for the configuration 50 train and 10 test

the velocity variable is removed obtains error percentages quite high, what confirms this variable importance.

On the other hand, the research determines that the influence of the mass on learning is not very relevant. Therefore and taking into account its numeric simulation cost, the network without this variable is considered the most efficient.

The network with the worst results is the one with less information, that is to say, with less input variables. However, it is advisable to highlight that the network without thickness and mass variables has quite acceptable results taking into account the little information it receives in comparison with the rest.

The knowledge acquired as a result of this research, can be spread out to other fields of great practical interest. Among these, the design of structural components of energy absorption stands out, being of great relevance in the field of passive safety vehicle.

Acknowledgements

We are grateful to Ramon Zaera and David Fernandez, who helped us in the development of this investigation.

References

[1] K. Swingler, *Applying Neural Networks: A Practical Guide.* Morgan Kaufmann, 1996.

[2] Z. Waszczyszyn and L. Ziemianski, “Neural networks in mechanics of structures and materials. new results and prospects of applications,” *Computers & Structures*, vol. 79 IS - 22-25, pp. 2261 EP– 2276, 2001.

[3] R. Ince, “Prediction of fracture parameters of concrete by artificial neural networks,” *Engineering Fracture Mechanics*, vol. 71, no. 15, pp. 2143–2159, 2004.

[4] S. Liu, J. Huang, J. Sung, and C. Lee, “Detection of cracks using neural networks and computational mechanics,” *Computer methods in applied mechanics and engineering (Comput. methods appl. mech. eng.)*, vol. 191, no. 25-26, pp. 2831–2845, 2002.

[5] W. Carpenter and J. Barthelemy, “A comparison of polynomial approximations and artificial neural nets as response surfaces,” *Structural and multidisciplinary optimization*, vol. 5, no. 3, p. 166, 1993.

[6] P. Hajela and E. Lee, “Topological optimization of rotorcraft subfloor structures for crash worthiness consideration.,” *Computers and Structures*, vol. 64, pp. 65–76, 1997.

[7] L. Lanzi, C. Bisagni, and S. Ricci, “Neural network systems to reproduce crash behavior of structural components,” *Computers structures*, vol. 82, no. 1, p. 93, 2004.

[8] A. Garcia, B. Ruiz, D. Fernandez, and R. Zaera, “Prediction of the response under impact of steel armours using a multilayer perceptron,” *Neural Computing & Applications*, 2006.

[9] C. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, USA, 1996.

[10] K. Hornik and M. Stinchcombe, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[11] R. Lippmann, “An introduction to computing with neural nets,” *ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine]*, vol. 4, no. 2, pp. 4–22, 1987.

[12] P. Isasi and I. Galvan, *Redes de neuronas artificiales: un enfoque practico.* Pearson Prentice Hal, 2004.

[13] L. Tarassenko, *A guide to neural computing applications.* Arnol / NCAF, 1998.

[14] U. ABAQUS Inc., Richmond, “Abaqus/explicit v6.4 users manual,” 2003.

[15] M. Gevrey, I. Dimopoulos, and S. Lek, “Review and comparison of methods to study the contribution of variables in artificial neural network models,” *Ecological Modelling*, vol. 160, pp. 249–264(16), 2003.