TB04

19th International Symposium of Ballistics, 7–11 May 2001, Interlaken, Switzerland

APPROXIMATING THE BALLISTIC PENETRATION FUNCTION OF A JET IN A MULTI-CASSETTE TARGET BY THE USE OF NEURAL NETWORKS

Eyal Gruss[1] and Eitan Hirsch[2]

[1] IDF, Mil. P.O.B. 01154, Israel (e-mail: eyal_gruss@yahoo.com)
[2] IDF, Mil. P.O.B. 02128, Israel (e-mail: eitanhir@isdn.net.il)

We present a novel artificial neural network architecture for modeling the ballistic penetration of a single shaped charge jet in a realistic multi-cassette target. After training on a large enough database, the model can estimate the expected residual penetration in different ballistic setups. In our model, a neural network is trained to model the ballistic action of a single generic cassette. Such cassette-networks are linked together to model the ballistic action of the multi-cassette target. We call this Multi-Cassette Analysis (MCA). Here we assume that the residual penetration after one cassette is the reference penetration of the residual jet for the succeeding cassette. An appropriate training method was developed for the MCA architecture, and preliminary results are presented. We hope that such a model may assist armor developers in making optimizations of armor configurations and armor materials, and constitute an efficient armor design tool.

INTRODUCTION

It is a difficult problem to model the ballistic performance of multi-cassette armor. Different methods are used to overcome this difficulty, including semi-analytical models [1], finite-difference numerical simulations [2] and of course experiments. Semi-analytical models are usually either over-simplistic or require parameter calibration. Finite-difference simulations require many parameter calibrations, cost considerable computational resources and take a lot of time. Experiments are expensive and destructive. Because of these disadvantages, the above methods are not efficient for optimization, for example, where many possible armor configurations must be considered, due to the large number of relevant parameters. Given an existing experimental database, a preferable solution may be to perform an intelligent regression on the empirical data.

Artificial neural networks are a common tool for performing non-linear regression or function approximation, especially when the parametric form of the function is unknown and when the number of parameters is large [3]. There are for example, interesting applications of neural networks in the field of material engineering [4] [5], but we have found

no references considering terminal ballistics. It would be nice if such a tool could be used to provide armor developers with an approximation of the residual penetration, P_{RES}, as a function of the ballistic experiment setup parameters:

$$P_{RES} = f(\text{warhead parameters, cassettes parameters, setup geometry}) \quad (1)$$

Finding a minimum of this function for a given warhead, under some practical constrains, may be termed "the terminal ballistics problem" (regarding armor developers). We find the naive use of a simple neural network for modeling the full ballistic problem to be ineffective. Instead, we propose a new *Multi-Cassette Analysis (MCA)* architecture that is tailored to our needs. This model is capable of estimating the residual penetration of a single shaped charge jet after penetrating a multi-cassette target, in a typical setup as shown in Fig. 1.

Figure 1: A typical ballistic experiment setup of a single shaped charge warhead against a multi-cassette target.

MODEL ARCHITECTURE

In our architecture, a three-layer feed-forward neural network models the ballistic action of a single generic cassette. This kind of network consists of an input layer, a hidden layer and an output layer. The layers are connected by weights that are the free parameters of the model. At the nodes of the hidden and output layers, a nonlinear sigmoid function is activated on the weighted sum of the previous layer's nodes. The network is trained by adjusting the weights, so that its output approaches a specified desired value. The cassette-network (Fig. 2) is fed with the appropriate warhead, cassette and geometry parameters, and the network is trained to yield a value close to the expected residual penetration after the cassette. A number of identical cassette-networks are linked together to constitute the MCA architecture, which models the ballistic action of a whole multi-cassette target (Fig. 3). The first cassette is fed with the warhead reference penetration, and the residual penetration after one cassette is used as the reference penetration of the residual jet fed to the succeeding cassette. Other than that, the cassettes are assumed to be independent. The model is trained to yield the experimentally measured overall residual penetration for a large number of training examples. Our model is based on an assumption that it is sufficient to pass forward only one historic parameter of the jet, which we refer to as parameter number one, in order to obtain a satisfactory description of the multi-cassette penetration.

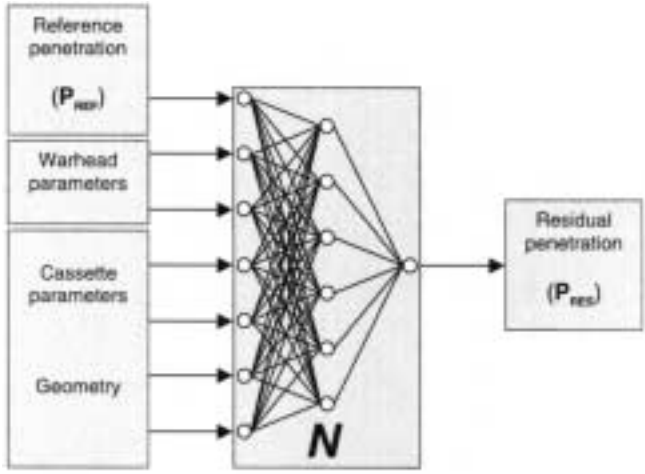

Figure 2: The generic cassette-network.

Our approach of modeling the cassettes themselves by using a separate (though identical) network for each cassette has several advantages. First, the number of free parameters of the model is reduced, allowing the use of a smaller database for training the model, or alternatively to increase the number of input parameters. Second, this realization of the physical covariance of cassette action allows the investigation and understanding of the action of a single cassette, and thus the description of armor composed of an arbitrary number of different cassettes. This is similar to the shared-weights approach used in [3] (regarding the T-C recognition problem). Finally, the MCA approach allows the incorporation of physical constraints between cassettes. In particular, we constrain the residual penetration after the cassettes to be monotonically declining and independent of succeeding cassettes, making the model more physically robust.

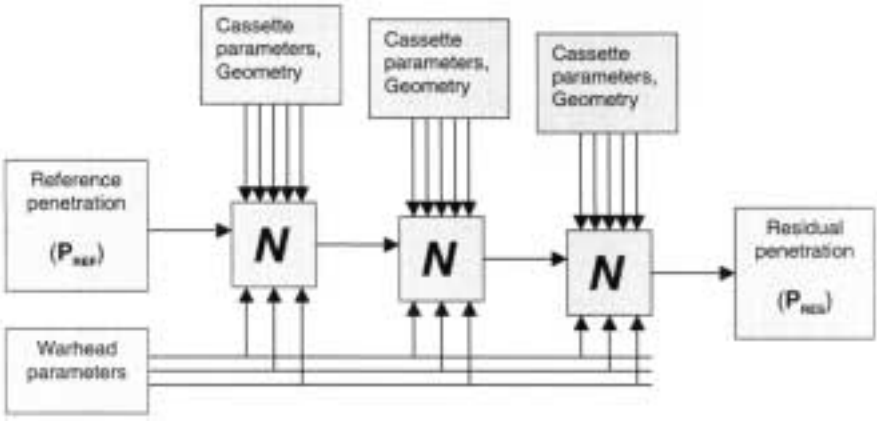

Figure 3: The MCA architecture. Identical neural networks, represented by the "*N*"-s, are used to model the different cassettes of the target.

PHYSICAL DATA PREPROCESSING

We consider armor in which each cassette is made out of two metal plates and some intermediate layer plate. In Table 1 are the physical parameters chosen to be the input parameters for the model. These were identified as the main parameters that influence the ballistic phenomena. It is important to choose parameters that are continues physical variables and not discrete categories. Besides these parameters, we have for every shot in the database the overall residual penetration measured in the experiment. We want to train our model to yield this output for the appropriate input, up to some given tolerance. We would like to work with a realistic tolerance, close to the empirical tolerance, in order not to make the training process too hard and not to have over-training (learning of the noise). With this intention, we chose the relevant tolerances as certain percentages of the difference between the warhead reference penetration and the overall residual penetration. We used a different percentage value for shots of large warheads (reference > 350 mm) and of small warheads (100–200 mm) due to the different variances in their results. Where the data included several shots with identical input parameters, the residual penetrations were averaged (without reducing the number of these shots' occurrences), and the tolerance was set to 120% of the maximum deviation from this average. As the warhead reference penetration at the actual standoff was not always known, we have used the reference penetration at some known standoff, forcing the model to learn the warhead's standoff curve as well. This is not ideal, however the model has enough input data to take this deviation into account.

Table 1: Input parameters for the cassette-network.

Category	Index	Input parameter
Parameter number one	1	Reference penetration
Warhead parameters	2	Warhead reference penetration
	3	Reference standoff from virtual origin
	4	Jet tip velocity
	5	Jet tail velocity
	6	Jet Breakup time
	7	Radius of particulated jet
Cassette parameters	8	1st plate thickness
	9	1st plate metal density
	10	1st plate metal yield strength
	11	2nd plate thickness
	12	2nd plate material density
	13	2nd plate material parameter #1 (application dependant)
	14	2nd plate material parameter #2 (application dependant)
	15	3rd plate thickness
	16	3rd plate metal density
	17	3rd plate metal yield strength
Geometry	18	Distance of cassette from virtual origin
	19	Angle of cassette
	20	Air gap after cassette

Except for parameter number one, the input parameters of the training examples were normalized to have zero mean and variance one. Parameter number one and the expected residual penetration, which are not known for the intermediate cassettes, were scaled to the range [–1, 1]. Principal component analysis may be applied to the normalized parameter space, in order to find the linearly independent dimensions of greatest variance. Then, dimensions of small variance can be neglected, reducing the original 20 input parameters to 13–14 effective input parameters that take account of about 98% of the variance in the data.

TRAINING SPECIFICS

A generic cassette-network is trained by back-propagation implemented with an incremental gradient decent scheme. The cost or error function is defined as

$$\text{error} = \tfrac{1}{2}(\text{output} - \text{target})^2 \tag{2}$$

where output is the network output (as a function of the inputs and weights) and target is the desired output, which is in our case the expected residual penetration. Weight adjustments are made in order to minimize this error, meaning to bring the output close to the target. In the case of gradient decent, the adjustments are made in a direction opposite to the gradient of the error function:

$$\text{weight}_{\text{new}} = \text{weight}_{\text{old}} - \eta \frac{\partial \text{error}}{\partial \text{weight}} \tag{3}$$

where η is a given learning rate. In the incremental method, the network weights are adjusted after each training example is fed to the network. Up till now everything is standard ([6] is a good reference for efficient implementation). When considering multi-cassette armor, however, the question arises how a number of linked networks can be trained using data from a single example. The main problem is determining the targets of the cassette-networks that are not the last cassette, because experimentally, the residual penetration is measured after the last cassette only. One approach to overcome the above problem would be to start with targets initialized by, say linear estimation, and make adjustments to the intermediate weights based on the overall error gradient, as for example:

$$\text{target}_{\text{new}} = \text{target}_{\text{old}} - \frac{\partial \text{overall error}}{\partial \text{output}} \tag{4}$$

where the gradient is computed at the intermediate cassette-network output. Practically, one can look at our linked networks, as a one large multi-layer network with many disconnected contacts, and make adjustments based on the overall error depending on the last output and target only. Therefore, an alternative approach to (4) would be to calculate all adjustments by regular back-propagation gradient decent (3) using the overall error, as if we were indeed dealing with one large network, with no need for intermediate targets. Undesirably, this would prevent us the possibility to constrain the training of intermediate

cassette-networks. However, it can be shown that the exact same adjustments will take place in our MCA architecture if the intermediate targets are set as

$$\text{target}_{\text{new}} = \text{output} - \frac{\partial \text{overall error}}{\partial \text{output}} \tag{5}$$

where again, output refers to the intermediate cassette-network output. This is since (5) causes the gradient of the intermediate cassette-network error to be equal to the overall error gradient. Interestingly, this method was found superior to different variants of adjusting intermediate targets by the former approach. Either way, training starts with the last cassette-network, for which the residual penetration is known, moving backwards. An intermediate target is evaluated as explained above, and is delimited to be smaller than both the preceding cassette-network output and the warhead reference penetration, and larger than both the succeeding cassette-network output and target. Then the respective cassette-network is trained by regular gradient decent (3) using the cassette-network error. Since we do not want the training process of one example to be iterative, we set the generic cassette-network to be the average of the adjusted cassette-networks of the current example. This may be viewed as a shared-weights approach or alternatively as a mini-batch session in the incremental scheme. A further improvement is made by delimiting also the actual cassette-network output to be smaller than both the preceding cassette-network output and the warhead reference penetration, and larger than the overall residual penetration for training examples, or than zero for test examples or new examples.

The use of an incremental method rather than a batch method is a requirement of the stochastic or adaptive nature of the data, which is due to the changes of intermediate input and target values. In addition, it has the advantages of lower memory requirements, better efficiency when there is a high amonnt of redundancy in the database, and its stochastic nature can help prevent convergence to a local, rather than global, minimum. Separate learning rates are used for input-to-hidden and for hidden-to-output weights, and they are automatically adapted by the method of [7] which proved to be very effective in dealing with the changing data. It was found very important to limit the minimum allowed learning rate. Weight decay in the method of [8] is used in order to maximize the generalization capability of the model. A disadvantage of our approach is that our model may require fine parameter tuning in order to achieve convergence of the learning process.

A training error threshold is set and it is gradually lowered each time that all training examples' outputs are within its limits. The model is trained for examples with an error larger then the current threshold or for which the intermediate output values are not monotonically declining. By this, more emphasis is put on examples harder for learning, helping the process to converge. However, a prior check should be done to exclude irregular training examples that are not in consent with the rest of the database and could deem convergence impossible. This can be done by counting for which examples the model does not yield satisfactory output a large number of times relative to the number of training epochs. The training examples are shuffled to a random order at the beginning of each training epoch. Training is stopped when the model reaches the specified training tolerance for all training examples, and when for each of them the intermediate output values are monotonically declining.

PRELIMINARY RESULTS

The model was implemented in C++ code. We studied an empirical database of 400 shots of different warheads against armor consisting of one to five cassettes of different configurations. 10 irregular shots were excluded from the database after found difficult for learning. Of the remaining, a validation set of 7%–10% test shots, which do not participate in the training process, were chosen randomly and were put aside. The remaining training shots were learned up to a tolerance of 6% for large warheads and 12% for small warheads. The residual penetrations of the test shots were accurately predicted within a tolerance of 7.5% for large warheads and 15% for small warheads. This achievement was repeated many times.

In conclusion, our feasibility test has yielded satisfactory results that confirm the model's generalization capability. It is clear that the quality of prediction depends on the degree of difference of the examples to be predicted from the training database, and thus depends on the extent that the database covers the ranges of parameters of interest. We therefore hope that aher training the model on a large database it may serve as an efficient armor design tool.

REFERENCES

1. N. Barnea, N. Sela and M. Ravid, "An Analytical Model for Shaped-Charge Jet Interaction with Reactive Armour and Residual Penetration", *Proceedings of the 14th International Symposium on Ballistics*, 1993.
2. K. Thoma, D. Vinckier, J. Kiermeir, U. Deisenroth and W. Fucke, "Shaped Charge Jet Interaction with Highly Effective Passive Sandwich Systems – Experiments and Analysis", *Propellants, Explosives, Pyrotechnics 18*, 275–281, 1993.
3. D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning Internal Representations by Error Propagation" in D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing – Explorations in the Microstructure of Cognition, Volume 1: Foundations*, The MIT Press, Cambridge, MA, 318–362, 1986.
4. K. J. Cios, G. Y. Baaklini, A. Vary and R. E. Tija, "Radial Basis Function Network Learns Ceramic Processing and Predicts Related Strength and Density", *Journal of Testing and Evaluation 22*, 343–350, 1994.
5. D. S. Chen, D. S. H. Wong and C. Y. Chen, "Neural Network Correlations of Detonation Properties of High Energy Explosives", *Propellants, Explosives, Pyrotechnics 23*, 296–300, 1998.
6. Y. LeCun, L. Bottou, G. B. Orr and K. R. Muller, "Efficient BackProp" in G. B. Orr and K. R. Muller, *Neural Networks: tricks of the trade*, Springer, 1998.
7. N. Murata, K. R. Muller, A. Ziehe and S. Amari, "Adaptive On-line Learning in Changing Environments" in M. C. Mozer, M. 1. Jordan and T. Petsche (eds.), *Advances in Neural Information Processing Systems 9*, The MIT Press, Cambridge, MA, 599–605, 1997.
8. A. S. Weigend, D. E. Rumelhart and B. A. Huberman, "Generalization by Weight-Elimination with Application to Forecasting" in R. P. Lippmann, J. E. Moody, and D. S. Touretzky (eds.), *Advances in Neural Information Processing Systems 3*, Morgan Kaufmann Publishers, San Mateo, CA, 875–882, 1991.