SIMULATION OF THE FLIGHT DISTANCES OF JAVELINS BASED ON A NEURAL NETWORK APPROACH

Klaus D. Maier, Veit Wank, Friedrich-Schiller-Universität, Jena, Germany,
Klaus Bartonietz,
Olympiastützpunkt Rheinland-Pfalz/Saarland, Saarbrücken, Germany

KEY WORDS: javelin throwing, neural networks, flight simulation

INTRODUCTION: The flight distances of javelins are determined by the release parameters as well as by the forces that act on the javelin during flight. The former are under the control of the thrower, whereas the latter are not. The flight phase of the javelin has been under investigation by a number of researchers using engineering approaches to model the flight phase (e.g., Best, R.J., Bartlett, R.M. & Sawyer R.A., 1995; Hubbard, M., 1984). The objective is to find the optimal javelin release parameters for achieving a maximal flight distance. The measurement of release parameters is not very precise. Determining the wind influence during flight is difficult. Therefore the models of javelin flight are based on distorted data. Artificial neural networks (NNs; Haykin, S., 1994; Jain, A.K. & Mao, J., 1996) are a powerful information processing technology that allows us to construct an input-output model of a problem by learning from examples. They are able to generalize, i.e. to produce reasonable outputs for inputs that have not been encountered during learning. NNs handle fuzzy, faulty and imprecise data well. Hence an alternative model to simulate flight distance can be designed and might offer advantages compared to the existing models.

METHODS: Release parameters have been measured using three dimensional film and video analysis. Relevant parameters were determined: the angle of release α, the angle of attack β (seen from the side), the angle of side attack φ (seen from above), as well as the velocity of release v (Figure 1). The overall flight distance d was measured as the distance between the throwing line and the athlete's hand at the point of release d_0 plus the distance between the line and the point of touch down of the javelin. Other parameters such as javelin brand, release height h_0, wind speed, etc., were not considered in the model.

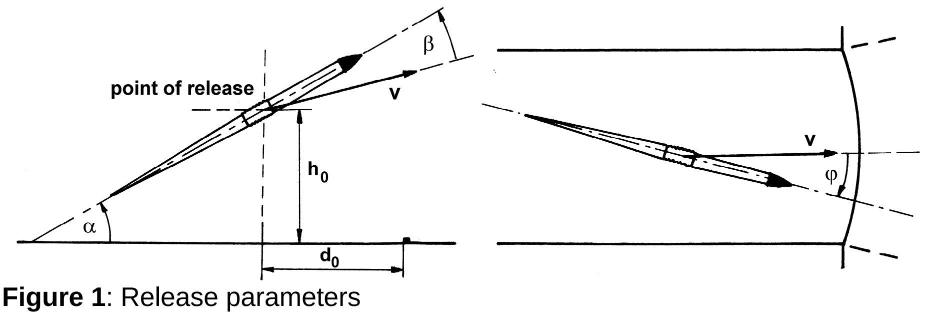

Figure 1: Release parameters

So-called multi-layer-perceptron neural networks (MLPs; Jain, A.K. & Mao, J., 1996) were used to construct a model with the release parameters as inputs and the overall distance as output (Figure). MLPs are made up of 'neurons' that have a number of inputs and generate an output using nonlinearity. Neurons in a MLP can be divided into input, output and neurons that are neither of the two – so-called hidden neurons. The whole structure is grouped in layers of neurons, i.e., input layer, output layer and a number of hidden layers of neurons that can be seen as parallel processing units. The hidden layers enable the MLP to learn complex tasks by extracting properties from the input patterns. One neuron in a layer is connected with all neurons in the following layer using connections with weights. The input of a neuron consists of the outputs of the previous layer multiplied by each of the connection weights. During training the weight of the connections is adjusted by means of the error back-propagation algorithm (BP, Rummelhart, D.E., Hinton, G.E. & Williams R.J., 1986). BP can be seen as a gradient-descent method to find a good weight configuration by minimizing an error cost function by repeatedly presenting an input-output pattern starting from randomly distributed weights. Once a MLP is trained it has input data fed into it and generates an appropriate output (retrieve). An important issue is the ability to generalize: MLPs can be 'overtrained' using BP – this results in closely matching outputs for trained data patterns but poor outcome for untrained patterns. To avoid this the cross-validation technique (CV; Stone, M., 1974) is used to choose good weight configurations after completing the BP: The original set of training data was split into three groups: The training data set (TDS) with 37 data patterns and two CV-sets (CVS): the validation data set (VDS) with two patterns and a single pattern to evaluate the performance (EPS) of the MLP.

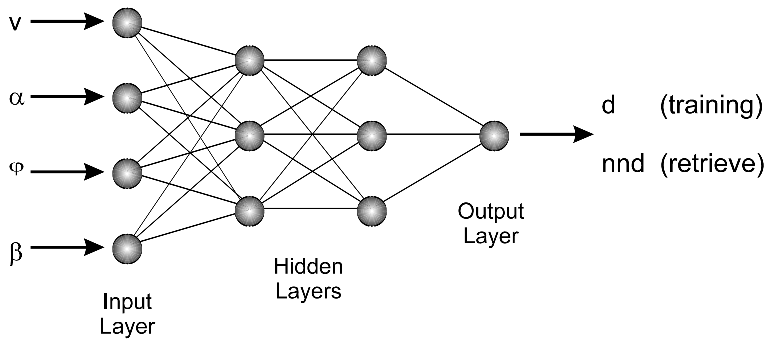

Figure 2: Structure of the MLP. Inputs are the release parameters, outputs are the overall distances d when training and the simulated distance nnd when retrieving.

RESULTS: We experimented with different MLP architectures using one or two hidden layers with between three and 30 neurons in each layer. A small setup that showed good results consisted of four input neurons (matching the number of input data), one output neuron (the total flight distance) and two hidden layers with three neurons each. The learning algorithm was stopped when the summed square error for all TDS dropped below 1m.

The VDS was used to test the generalization performance of the MLP – the predicted distance nnd had to be in a range of ±5 percent difference to the original d in order for the generalization criterion to be matched. The final prediction quality was tested on the EPS. We used four main sets (see Table) with each randomly chosen CVS from the 40 original data patterns, the remaining patterns made up of TDS.

Table 1: The 4 main CV sets. The first number in the set defines the main set and the second is the number of the pattern used for defining different VDS setups.

Set	v [m/s]	α [degrees]	β [degrees]	φ [degrees]	Distance [m]
1 - 1	28.0	34	35	14	83.34
1 - 2	29.2	37	41	14	87.58
1 - 3	24.0	35	38	8	59.50
2 - 1	28.5	34	36	14	80.99
2 - 2	23.9	34	39	8	58.10
2 - 3	22.5	35.5	36	1	58.10
3 - 1	26.4	25	30	14	68.86
3 - 2	22.4	40	47	11	55.25
3 - 3	24.5	31	35	4,5	66.16
4 - 1	21.3	36	40	11	48.00
4 - 2	22.6	40	49	12	54.80
4 - 3	27.6	35	41	7	78.08

For each of the four main sets (MS) we introduced three subsets with all variations of the three CVS data patterns for the VDS and the EPS (see Table 2 for MS one, MSs two, three and four are set up analogously).

Table 2: Subset arrangement for training and performance evaluation of MS one.

Subset	VDS	EPS
1	1-2, 1-3	1-1
2	1-1, 1-3	1-2
3	1-1, 1-2	1-3

Results for EPS with ten learning trials using this setup are displayed in Table .

Table 3: Prediction result for all VD setups.

EPS	nnd 1 [m]	nnd 2 [m]	nnd 3 [m]	nnd 4 [m]	nnd 5 [m]	nnd 6 [m]	nnd 7 [m]	nnd 8 [m]	nnd 9 [m]	nnd 10 [m]
1-1	81,57	80,42	79,11	79,33	80,37	78,53	80,26	81,61	78,28	80,48
1-2	82,51	84,60	89,99	84,40	84,50	89,21	88,89	83,58	90,22	83,10
1-3	59,94	57,69	60,26	62,19	56,61	62,94	56,73	57,24	60,13	57,91
2-1	80,67	80,87	80,65	80,67	79,86	80,08	80,56	80,02	80,68	79,17
2-2	61,04	61,91	60,76	60,77	61,38	61,01	61,17	61,76	60,59	61,02
2-3	56,29	56,01	56,05	56,15	55,65	56,29	56,07	56,24	55,92	55,67
3-1	70,97	71,65	71,36	71,06	71,64	69,07	71,63	70,84	70,69	70,95
3-2	54,25	54,81	55,37	54,94	54,60	54,59	55,60	55,33	55,26	54,58
3-3	63,42	64,15	65,06	63,74	63,62	64,86	64,14	63,55	63,81	63,19
4-1	50,83	50,73	50,77	50,71	50,80	50,91	51,04	49,03	50,72	49,07
4-2	55,72	55,38	55,99	55,57	56,00	56,42	55,33	55,58	55,91	55,93
4-3	76,18	76,44	76,50	75,94	76,21	77,74	76,21	78,87	76,55	76,41

The mean of the simulated distances, standard deviation for ten trials, as well as the difference between d and the mean of the simulated distance in percent are shown in Table 4.

Table 4: Mean of simulated distances (Mn. in m), standard deviation of Mn. (Stv. in m) and difference between simulated and real distance (∆% in percent).

EPS	1-1	1-2	1-3	2-1	2-2	2-3	3-1	3-2	3-3	4-1	4-2	4-3
Mn.	79,9	86,1	59,2	80,3	61,1	56,0	71,0	54,9	64,0	50,5	55,8	76,7
Sdv.	1,16	3,08	2,26	0,53	0,43	0,23	0,76	0,44	0,61	0,75	0,33	0,90
∆%	-4,0	-1,7	-0,6	-0,8	5,2	-3,6	3,1	-0,6	-3,3	5,1	1,8	-1,8

The simulated flight distances ndd are accurate up to about five percent. Variation between different ndd is small and different learning trials therefore show similar results.

CONCLUSIONS: We demonstrated that a neural network based approach is a suitable instrument for simulation of the flight distance, even though it uses only a small number of parameters. This tool can assist coaching and provides an alternative to other models. Results are encouragingly accurate, considering the number of training sets. To further improve the quality of prediction more data sets have to be used for training the MLPs. Future work will include recording more data, as well as studies on optimal javelin release parameters.

We would like to thank Prof. R. Blickhan for providing the facilities that made this work possible. Prof. C. Beckstein is also to be thanked for helpful advice on neural networks and stimulating discussions.

REFERENCES:

Best, R. J., Bartlett, R. M., Sawyer R. A. (1995). Optimal Javelin Release. *Journal of Applied Biomechanics* **11**, 371-394.
Haykin, S. (1994). Neural Networks – A Comprehensive Foundation. Englewood Cliffs, N.J.: Macmillan Publishing Company.
Hubbard, M. (1984). Optimal Javelin Trajectories. *Journal of Biomechanics* **17**, 777-787.
Jain, A. K., Mao, J. (1996). Artificial Neural Networks: A Tutorial. *Computer* **3**, 31-45.
Rummelhart, D. E., Hinton, G. E., Williams R. J. (1986). Learning Representations by Back-Propagating Errors. *Nature* **323**, 533-536.
Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society* **B36**, 111-133.