J. Phys. G: Nucl. Part. Phys. 20 (1994) L21–L26. Printed in the UK

LETTER TO THE EDITOR

Neural networks for impact parameter determination*

S A Bass†‡, A Bischoff†, C Hartnack†§, J A Maruhn†, J Reinhardt†, H Stöcker† and W Greiner†

† Institut für Theoretische Physik der Universität, Postfach 111932, 60054 Frankfurt am Main, Federal Republic of Germany

‡ GSI Darmstadt, Postfach 110552, 64220 Darmstadt, Federal Republic of Germany

§ Laboratoire de Physique Nucléaire, Nantes, France

Received 8 November 1993

Abstract. Accurate impact parameter determination in a heavy-ion collision is crucial for almost all further analysis. We investigate the capabilities of an artificial neural network in that respect. First results show that the neural network is capable of improving the accuracy of the impact parameter determination based on observables such as the flow angle, the average directed in-plane transverse momentum and the difference between transverse and longitudinal momenta. However, further investigations are necessary to discover the full potential of the neural network approach.

The physics of heavy-ion collisions is motivated by the idea to learn something about the properties of hot and dense nuclear matter [1–8]. In order to investigate highly compressed nuclear matter in heavy-ion collisions it is important to select only the most central collisions. On the other hand, recently discovered new phenomena such as pionic bounce-off and squeeze-out are only observed in semiperipheral collisions. Therefore the proper determination of the impact parameter in a heavy-ion collision is crucial to almost all further analysis. There have been various approaches towards the determination of the impact parameter, most of which are either based on the mean particle multiplicity or on a transverse momentum analysis (directivity-cut). However, all of these methods have one thing in common: they tend to break down for impact parameters smaller than 2 fermi. In general the accuracy is about ± 1 to ± 1.5 fermi.

In this contribution, an artificial neural network has been used to determine the impact parameter. For a proper analysis of the network's performance, the analysed heavy-ion collisions have to be supplied by a theoretical event-generator rather than by experiment. Otherwise it would be impossible to compare the network output with a target value for the impact parameter of the heavy-ion collision. For our investigations, we applied an extension of the quantum molecular dynamics model (QMD) [9–11], which explicitly incorporates isospin and pion production via the delta resonance (IQMD) [12–14]. In the QMD model the nucleons are represented by Gaussian-shaped density distributions. They are initialized in a sphere of a radius $R = 1.14A^{1/3}$ fm, according to the liquid-drop model. Each nucleon is supposed to occupy a volume of h^3, so that the phase space is uniformly filled. The initial momenta are randomly chosen between 0 and the local Thomas–Fermi momentum. The A_P and A_T nucleons interact via two- and three-body Skyrme forces, a Yukawa potential,

* Supported by GSI, BMFT and DFG.

0954-3899/94/010021+06$07.50

momentum-dependent interactions, a symmetry potential (to achieve a correct distribution of protons and neutrons in the nucleus) and explicit Coulomb forces between the Z_P and Z_T protons. They are propagated according to Hamilton's equations of motion. Hard N–N collisions are included by employing the collision term of the well known VUU/BUU equation [5, 15–18]. The collisions are done stochastically, in a similar way as in the cascade models [19, 20]. In addition, the Pauli blocking (for the final state) is taken into account by regarding the phase-space densities in the final states of a two-body collision.

As input for the neural net we use the flow angle ϑ_{flow}, the average directed in-plane transverse momentum $p_{x,\text{dir}}$, and an observable Q_{ZZ} which compares transverse with longitudinal momentum. The flow angle can be extracted from the flow-tensor

$$F_{ij} = \sum_{n=1}^{A_T+A_P} p_i(n)p_j(n) \qquad i,j = x,y,z$$

using a sphericity analysis [21].

The average directed in-plane transverse momentum $p_{x,\text{dir}}$ is defined as

$$p_{x,\text{dir}} = \frac{\sum_{i=1}^{A_T+A_P} p_x(i)\text{sgn}(Y(i) - Y_{CM})}{A_T + A_P}$$

and Q_{ZZ} can be calculated as

$$Q_{ZZ} = 3P_z^2 - P^2 = 2P_z^2 - P_x^2 - P_y^2 \qquad P_i = \sum_{n=1}^{A_1+A_P} p_i(n) \qquad i = x,y,z.$$

The impact parameter dependence of these observables has been well established [14].

We now sketch the neural network algorithm used, a standard feedforward two-layer perceptron trained by error-backpropagation [22, 23]. The network consists of a single 'hidden' layer of nonlinear units receiving inputs from the applied data vector (e.g. $(\vartheta_{\text{flow}}, p_{x,\text{dir}}, Q_{ZZ})$) and transferring their signals to the output unit (we use a single linear output unit, whose continuous-valued output represents the impact parameter).

In our feedforward network, every unit (of the hidden or output layer) is connected to each unit of the preceding layer, performing a weighted sum over all input signals, and finally calculating its own signal by applying a 'squashing' function to the result:

$$S_j^h = \Theta(Z_j^h) = \Theta\left(\sum_k w_{jk}^h y_k\right)$$

for hidden units, and

$$S_i = \theta(Z_i) = \theta\left(\sum_j w_{ij} S_j^h\right)$$

for output units. (As squashing functions we use $\Theta(Z) = \tanh(Z)$, and $\theta(Z) = Z$, respectively.) The connection weight, between units j to unit i, is given by w_{ij}. y_k is a component of the data vector. For each unit, we include a connection to a constant signal, $S_0 \equiv 1$, which provides an activity threshold.

First, the network's weights are initialized with small random values. During training, for each learning pattern an output is produced and rated by the error function

$$E = \tfrac{1}{2}\sum_i (S_i^{\text{turget}} - S_i)^2$$

where S_i^{target} is the desired output. Successively for each pattern, the weights are updated according to a gradient descent in the weight space with respect to the error function,

$$\Delta w_{nl} = -g\frac{\partial E}{\partial w_{nl}} \qquad \Delta w_{lm}^{\text{h}} = -g\frac{\partial E}{\partial w_{lm}^{\text{h}}}$$

with g as a learning parameter. This leads to the learning rules

$$\Delta w_{nl} = g(S_n^{\text{target}} - S_n)\theta'(Z_n)S_l^{\text{h}} = g\delta_n S_l^{\text{h}}$$

using the definition

$$\delta_n = (S_n^{\text{target}} - S_n)\theta'(Z_n)$$

and

$$\Delta w_{lm}^{\text{h}} = g\sum_i \left(S_i^{\text{target}} - S_i\right)\theta'\left(\sum_j w_{ij}S_j^{\text{h}}\right)w_{il}\frac{\partial S_l^{\text{h}}}{\partial w_{lm}^{\text{h}}} = g\delta_l^{\text{h}} y_m$$

where

$$\delta_l^{\text{h}} = \sum_n \delta_n w_{nl}\Theta'(Z_l^{\text{h}}).$$

For a complete learning session, typically several hundred cycles through the entire training data are necessary. Training is stopped when the performance on a set of test data does not improve any further. In contrast to the extensive training phase, very little calculation time is needed for the application of a trained network. A trained network may even be transformed into electronic hardware, which would be faster than our computer simulation by orders of magnitude.

Two sets, each containing 2600 Au(1 *A* GeV)Au events, generated by the IQMD model [14], form the simulation data. The first data set contains a minimum bias calculation, thus giving a sample from the full impact parameter range, whereas the second one is limited to impact parameters smaller than three fermi. Both data sets have been subjected to an angular cut according to the acceptance of the FOPI spectrometer in its phase II setup [24]. For each impact parameter range, the network is provided with 10% of the data as learning samples and then it has to estimate the impact parameter for the remaining 90% of events. To avoid overfitting, we controlled the network's performance after each learning cycle by applying the full test data-set. The network used for the full input vector with three components consisted of five hidden units and one output neuron. In order to determine the amount of hidden units needed for the best network performance, the number of hidden units was increased from one to eight in single increments and the respective network performance was tested. No significant performance increase was achieved for more than five hidden units.

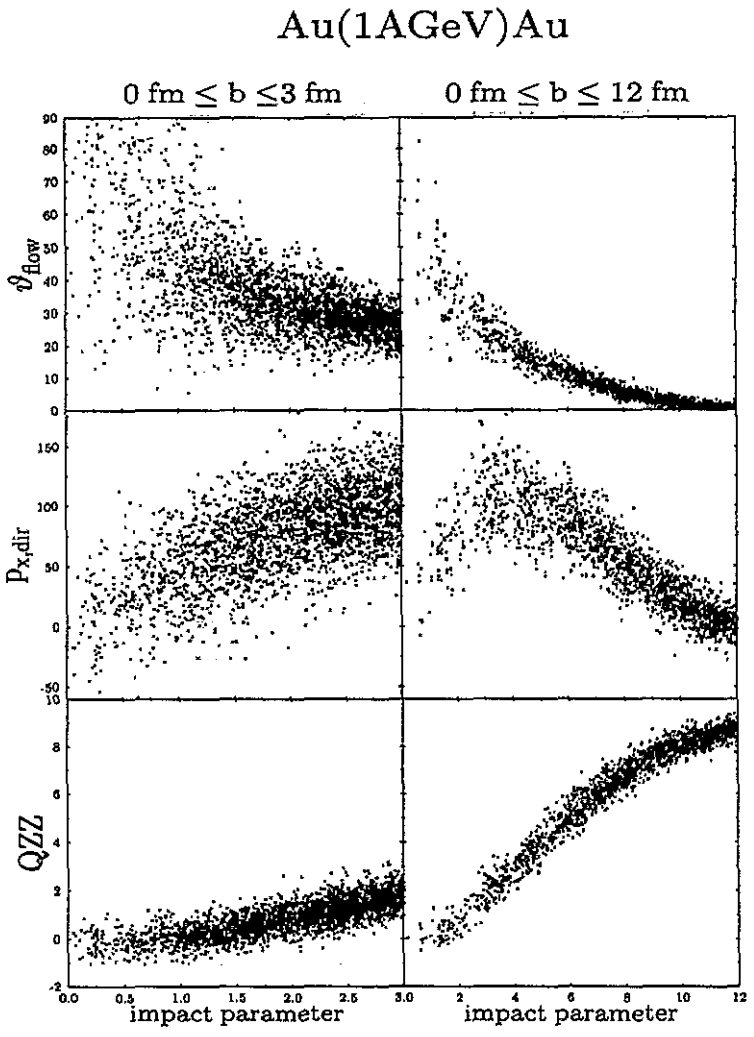

Figure 1. Dependence between impact parameter b and ϑ_{flow}, $p_{x,\text{dir}}$ and Q_{ZZ} for central and minimum bias Au(1 *A* GeV)Au events as calculated by the IQMD model. Each event is represented as a dot. Both the learning sample and the test sample are plotted.

Figure 1 shows scatter plots (each dot representing one event) describing the functional dependence of the impact parameter on either one of the three inputs, ϑ_{flow}, $p_{x,\text{dir}}$ and Q_{ZZ}, as calculated by the IQMD model. Using the minimum bias data set, the neural network output essentially produced a nonlinear fit to the impact parameter dependence of these inputs which is also intuitively apparent from the figure. In order to minimize errors due to statistical fluctuations in the events, it is, however, desirable to simultaneously fit the impact parameter as a function of ϑ_{flow}, $p_{x,\text{dir}}$ and Q_{ZZ}. This gives a trajectory in a three-dimensional phase space with the impact parameter as an evolution parameter. For standard fitting techniques this poses a more difficult problem, which we instead left again to the neural network. Table 1 shows the average absolute error (in fermi) between the impact parameter estimated by the neural network and the actual impact parameter (using the test data set) for the functions $b(\vartheta_{\text{flow}})$, $b(p_{x,\text{dir}})$, $b(Q_{ZZ})$ and $b(\vartheta_{\text{flow}}, p_{x,\text{dir}}, Q_{ZZ})$ as learned by the neural net.

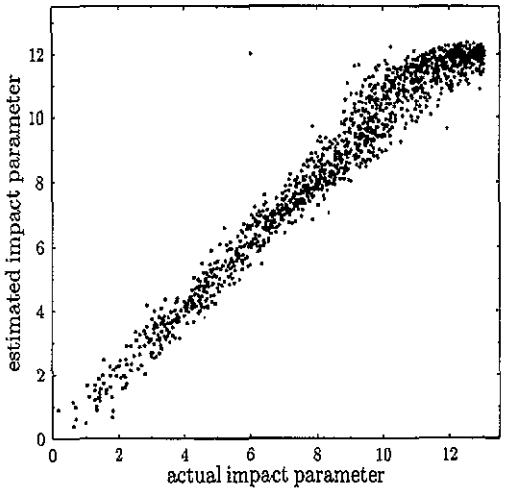

Figure 2. True impact parameter versus impact parameter predicted by the neural network for the minimum bias data-set using the full input vector. The best performance is achieved in the 3 to 9 fm impact parameter range.

Table 1. Average absolute error (in fermi) between the impact parameter b learned by the neural network and the actual impact parameter as a function of either ϑ_{flow}, $p_{x,\text{dir}}$ or Q_{ZZ} and as a function of ϑ_{flow}, $p_{x,\text{dir}}$ and Q_{ZZ} simultaneously.

Impact parameter	0 fm $\leqslant b \leqslant$ 3 fm	0 fm $\leqslant b \leqslant$ 12 fm
$\Delta b(\vartheta_{\text{flow}})$	0.43	0.62
$\Delta b(p_{x,\text{dir}})$	0.45	1.05
$\Delta b(Q_{ZZ})$	0.34	0.75
$\Delta b(\vartheta_{\text{flow}}, p_{x,\text{dir}}, Q_{ZZ})$	0.32	0.47

For the minimum bias data-set the neural network is able to extract a better fit using all three inputs than using only one of them. For central events this is not the case: a simple mapping of $b(Q_{ZZ})$ shows almost identical results. This becomes understandable looking at figure 1, where $b(Q_{ZZ})$ clearly shows the least fluctuations.

Figure 2 shows a graphical representation of the network performance on the minimum bias data-set using the full input vector. The true impact parameter is plotted versus the impact parameter estimated by the neural network. The best performance is achieved in the range between 3 fm and 9 fm. For very central and very peripheral collisions the width of the output deviations increases.

With the selected preprocessed event input, classical fitting techniques suffice for a good estimate of the impact parameter. Compared with previously used techniques [25], the suggested observables can improve the accuracy of the impact parameter determination. However, we surely have not used the neural network approach up to its full potential: a larger network will most likely be capable of handling raw event-data and recognizing patterns which are indistinguishable for classical types of analysis. In addition, more sophisticated network models (e.g. with higher order or optimized connectivity [26, 27]) could be used instead of our exploratory straightforward model to gain calculational efficiency. In the form of a VLSI chip it might even serve as a hardware event-trigger for the impact parameter.

References

[1] Scheid W, Ligensa R and Greiner W 1968 *Phys. Rev. Lett.* **21** 1479
[2] Scheid W, Müller H and Greiner W 1974 *Phys. Rev. Lett.* **32** 741
[3] Csernai L P and Kapusta J I 1986 *Phys. Rep.* **131** 225
[4] Stock R 1986 *Phys. Rep.* **135** 261
[5] Stöcker H and Greiner W 1986 *Phys. Rep.* **137** 277
[6] Clare R B and Strottman D 1986 *Phys. Rep.* **141** 179
[7] Schürmann B, Zwermann W and Malfliet R 1986 *Phys. Rep.* **147** 3
[8] Cassing W, Metag V, Mosel U and Niita K 1990 *Phys. Rep.* **188** 365
[9] Aichelin J and Stöcker H 1986 *Phys. Lett.* **176B** 14
[10] Aichelin J, Rosenhauer A, Peilert G, Stöcker H and Greiner W 1987 *Phys. Rev. Lett.* **58** 1926
[11] Aichelin J 1991 *Phys. Rep.* **202** 233
[12] Hartnack Ch, Stöcker H and Greiner W 1988 *Proc. Int. Workshop on Gross Properties of Nuclei and Nuclear Excitation XVI (Hirschegg, Kleinwalsertal, Austria)* ed H Feldmeier
[13] Hartnack C, Zhuxia L, Neise L, Peilert G, Rosenhauer A, Sorge H, Aichelin J, Stöcker H and Greiner W 1989 *Nucl. Phys.* A **495** 303
[14] Hartnack Ch 1993 *PhD thesis* (*GSI-Report* 93-5)
[15] Kruse H, Jacak B V and Stöcker H 1985 *Phys. Rev. Lett.* **54** 289
[16] Aichelin J and Bertsch G 1985 *Phys. Rev.* C **31** 1730
[17] Wolf G, Batko G, Cassing W, Mosel U, Niita K and Schäfer M 1990 *Nucl. Phys.* A **517** 615
[18] Li B A, Bauer W and Bertsch G F 1991 *Phys. Rev.* C **44** 2095
[19] Yariv Y and Frankel Z 1979 *Phys. Rev.* C **20** 2227
[20] Cugnon J 1980 *Phys. Rev.* C **22** 1885
[21] Danielewicz P and Odyniec G 1985 *Phys. Lett.* **157B** 146
[22] McClelland J L and Rumelhart D E (ed) 1986 *Parallel Distributed Processing* vols 1 and 2 (Cambridge, MA: MIT Press)
[23] See also Müller B and Reinhardt J 1990 *Neural Networks* (Berlin: Springer) and references therein
[24] Pelte D, Reisdorf W and Wienold T (for the FOPI collaboration) 1993 *GSI Nachrichten* 9-93 (unpublished) and private communications
[25] Grigorian Y I (for the FOPI-Collaboration) 1992 *GSI Scientific Report 1991* p 28, and various other contributions by the FOPI-Collaboration in the same report
[26] Bishcoff A and Schürmann B 1993 *Artificial Neural Networks* vol 3 (Amsterdam) (to appear)
[27] Hergert F, Finnoff W and Zimmermann H G 1993 *Neural Networks* (to be published)