EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN-ECP/95-6
March 1995

fu 9513

Neural Second-Level Trigger System Based on Calorimetry

J.M. Seixas¹⁾²⁾, L.P. Caloba¹⁾²⁾, M.N. Souza²⁾, A.L. Braga²⁾, A.P. Rodrigues²⁾

CERN LIBRARIES, GENEVA

CERN-ECP-95-06

Abstract

A second-level triggering system based on calorimetry is analyzed using neural networks. Calorimeter data in a LHC environment is obtained with Monte Carlo simulations and an algorithm for the first-level trigger operation is applied. The surviving events are then available as a 20x20 matrix information corresponding to the calorimeter towers in the region of interest. The dominant background for triggering on electrons is assumed to consist of QCD jets which passed the first-level trigger condition.

The main features of the calorimeter are extracted. Matrix information, shower deposition in concentric rings and tail weighting procedures are studied. The processed information is sent to a fully connected backpropagation neural network. In this analysis we also consider pileup effects of an average of 20 minimum bias events. The neural network based system achieved up to 99% electron efficiency with less than 9% of jets being misclassified as electrons.

(Submitted to Nuclear Instruments and Methods)

¹⁾ CERN, Geneva, Switzerland.

²⁾ COPPE/EE/UFRJ, Rio de Janeiro, Brazil

1 Introduction

Neural networks have proven to perform well as classifiers in many applications. In high-energy physics we find pattern recognition problems ranging from low-level experimental triggers to off-line analysis. For calorimeters, the identification of particles have been tried successfully with neural networks [1]. It is known that this technique can be even more powerful if the raw data is pre-processed and sent as input to the network. The pre-processing can be performed by a non-neural network system.

Neural network design comprises two phases. During the training phase, representative events are shown to the network whose parameters are changed according to a given rule (i.e. backpropagation [2]) and a given cost function is thereby minimized. The testing or production phase follows and tests the network's ability to generalize the learned event classification on a set of events which were never before presented to the network. In this phase, the network assigns some probability to the incoming event to belong to a given class.

The event selectivity is known to be quite critical for the LHC experiment, due to the high event rate of 40 MHz which has to be dramatically reduced by the triggering system. This process of triggering is expected to be implemented in different levels and each level deals with events that passed the conditions of the previous ones. For LHC, after a reduction by a first-level trigger, the event passing rate is foreseen to be 100 kHz.

For the second-level trigger, calorimeter information can be combined with various other subdetectors. The operation of this system can be split into two phases [3, 4]. The first, so called feature extraction phase, consists in converting raw front-end data into meaningful quantities (features) from the physics point of view. These features are obtained from restricted areas of each subdetector, the Regions of Interest (ROI), which are identified by the first-level trigger. The second phase combines the features extracted from the subdetectors into a global decision unit. The second-level trigger has to be quite fast and neural networks can be considered as a competitive technique in which speed, implementation and performance are concerned [5, 6].

The feature extraction problem for calorimeters is considered in this paper. The features to be extracted are found by searching for the best calorimeter variables one could send to the input nodes of a neural network discriminator capable to perform electron/jets separation based only on calorimeter information. For this purpose, Monte Carlo simulations have been used to generate events for the second-level operation [7]. A large sample of QCD jets and single electrons were generated using Pythia [8], both in the same pseudorapidity (η) range. The calorimeter has cells of size $0.02 \times 2\pi/300$ in $\eta \times \Phi$. The detector has electromagnetic (e.m.) and hadronic sections with cells of the same size and precisely aligned. In order to avoid excessive computing time, only barrel cells were considered.

On the sample data the first-level trigger algorithm that has been proposed for the ATLAS experiment [9] was applied and afterwards pileup added. An average of 20 minimum bias events were combined to simulate the high-luminosity operation of LHC. The first-level algorithm was then applied again. In this manner, two data sets were available for analysis, covering low (no pileup) and high-luminosity scenarios, with a total of 1057 (995 when pileup was added) jets and 1634 (1625) electrons. Each of the retained events is described by means of two matrices of 20×20 elements, corresponding to the e.m. and hadronic parts of the calorimeter in the ROI. Figure 1 shows how typical electron and jet events look like.

show any improvement.

As LHC will not start operation at full luminosity, it would be interesting to see how the performance of the neural discriminator deteriorates with increasing luminosity. Thus, we tested whether the neural network technique is robust for operating with pileup events after being trained with events having no pileup. For this purpose, we trained the network with 5 randomly selected boxes of the data set generated for low-luminosity operation and tested it using events with pileup. Again, 10 different tests were performed. The result obtained was a 97% electron efficiency with an error for jets of 15.5%. This is to be compared with 13.8% error on jets obtained before for the same electron efficiency and when the network was trained with pileup events. The result shows that the performance of the matrix method is relatively robust in terms of increasing luminosity. It should be noted that the high-luminosity scenario can be faced as shown before, training with events having pileup effects. Concerning implementation, this would mean to load different weight vectors to the network when the luminosity had been increased.

3 Ring Analysis

The concept of shower development in concentric rings has been explored in calorimetry [12]. Here this idea looks quite attractive as it can greatly reduce the dimension of the input space and, thus, provide a faster and more compact processing unit for the second-level trigger operation. Using an 11x11 total energy description for the ROI, we define 6 rings, taking the cell with maximum energy deposition as the first ring and building the other rings around this cell. Thus, the pre-processing required involves finding this central cell and performing the sums over the energy deposition in each ring.

We have explored two different ways of building rings. In Figure 3(a) we see rings built around the cell of maximum deposited energy. The first ring is composed by this single cell; the second comprises all cells around the central one and so on. The cells belonging to each ring are added to define the respective ring sum and this information is fed into the neural network. Figure 3(b) shows a similar way of defining ring sums, but now a circular ring structure is built by weighting accordingly the energy information of each cell.

The normalization for ring analysis is carried out in two ways. The first method takes the total energy (E_t) in the 11x11 region as the normalizing factor for all rings. A second method proved to work better and takes a sequential energy profile: the first ring is divided by E_t ; the second ring is divided by $E_t - E_{r1}$, where E_{r1} represents the total energy of the first ring (or central cell). From the third to the sixth ring, the normalizing factor taken is $E_t - (E_{r1} + E_{r2})$, E_{r2} being the total energy inside the second ring. Using this normalization scheme we weight each ring sum value and this reveals better the important contribution of the farthest rings to the discrimination process.

As the calorimeter has two sections, we could also explore ring information coming from e.m. and hadronic sections separately. In general, the hadronic section samples very few energy for events having passed the first-level trigger conditions. Different energy normalization factors were tested but we did not find any improvement in performance by feeding into the network this segmented information.

The analysis using rings can be improved by using variables coming from the first-level trigger algorithm. Define an e.m. isolation ring, E_{misol} , and a central cluster, $E_{tcluster}$, in a full ROI, as it is shown in Figure 4 [9]. Non-overlapping groups of 5x5 calorimeter cells are formed. The isolation ring comprises all cells along the edge of the 4x4 square

and a cluster region is defined to consist of two adjacent 5x5 cells not touching the edge of the 4x4 square. The cluster energy, $E_{tcluster}$, is taken as the maximum of the two 2x1 groupings. Both E_{misol} and $E_{tcluster}$ are normalized by the total energy in the subregion of interest. The information coming from these two variables is added to the ring information, forming an eight-dimensional input space.

Table 1 resumes results from the ring analysis. All features shown use a sequential profile normalization. Normalizing squared rings by the total energy gives $\sim 10\%$ deterioration in performance. Therefore, the sequential profile was chosen to be the standard normalization procedure. From the table, it's noticeable that we can achieve an extra rejection factor using circular rings (9.7% of jets misclassified against 10.5%, for the same electron efficiency), comparing their performance with squared ones. Finally, the use of $E_{tcluster}$ and E_{misol} combined with ring information allows an even better performance, that is comparable with the matrix method.

4 R_p Mapping

A quantity R_p

$$R_p = \frac{\sum_{i=1}^{121} E_i^\alpha r_i^\beta}{\sum_{i=1}^{121} E_i^\alpha} \quad (1)$$

is known to be a good measurement of the effective shower width [13]. Here, E_i represents the energy deposited in an individual cell i , r_i is the Euclidean distance between the center of this cell and the cell of maximum deposited energy and α, β are parameters which can be adjusted. Again, we take the 11x11 ROI description. A similar quantity has also been used as part of the non-neural algorithm referred to before [7] for the electron/jet discrimination. The values used there for α and β were 1 and 2, respectively.

In this section we develop some features which are extensions of this basic idea. The R_p algorithm is going to be generalized as a pre-processing method for the neural network system. Firstly, we have searched for the optimum value of the adjustable parameters in order to have the best performance in terms of electron/jet separation. To achieve this goal, we minimized the distance between the error function (for jets misclassified as electrons) obtained by the application of R_p mapping and a target error function (see Figure 5) [14]. There is a special case of the target curve when we take an ideal curve of null error. Then, the measured distance is just the norm of the error function of the R_p discriminator. Mathematically, we have

$$\|f\| = \sqrt{\sum_{i=1}^N [er(ef f_i) - ert(ef f_i)]^2} \quad (2)$$

where $er(.)$ and $ert(.)$ are the error functions for jets of the R_p and the target discriminators, respectively, and the parameter eff is the efficiency function for electrons.

The ideal target ($ert(.)=0$) was chosen and only high-efficiency values were used to compute this norm. In the range 90% to 99% of electron efficiency, we used a resolution of 0.5%, which corresponds to compute the norm of a vector of 19 components. A total of 400 sample values of the norm were obtained.

Figure 6(a) shows how the norm varies according to different values of α and β when we apply the R_p discriminator on the whole data sets we have (without pileup). The optimum value selection is $\alpha = 0.9$ and $\beta = 3.6$. Figure 7 shows the distributions for electrons and jets obtained with the R_p mapping using these parameter values. Using the

optimum parameter selection, 97% electron efficiency is achieved with 9% of the jets being misclassified as electrons. It can be concluded that taking the norm of the error function as a figure of merit for the determination of the parameter values works as a tuning in the overall performance of this discriminator, as the discrimination with $\alpha = 1.0$ and $\beta = 2.0$ (values used in [7]) misclassifies 12% of the jets under the same conditions.

In terms of pileup events, the optimum choice of α and β can be different from the one found for events without pileup. This is illustrated in Figure 6(b). The optimal parameter values in the case of pileup are $\alpha = 0.9$ and $\beta = 1.4$. Using these pileup optimum values, one achieves 97% electron efficiency with 13.8% of jets wrongly classified, when events with pileup are considered. If one takes the non-pileup optimal values, 19% of jets are misclassified for the same 97% electron efficiency and under pileup conditions. This result shows that the weighting profile for pileup events is slightly different and the best value for β in this case is smaller than the case where pileup is not considered.

The R_p function, which maps the input vector space onto the real line, can be generalized, so that a R_p n-dimensional function is defined:

$$R_p : X \rightarrow Y \quad (3)$$

$$(E_i, r_i) \rightarrow \frac{E_i^\alpha \cdot r_i^\beta}{\sum_{i=1}^{121} E_i^\alpha}$$

In this way, the R_p n-dimensional function maps each cell contents in an energy weighted profile, using the radius as a weighting parameter. This function can be used as data pre-processing for a neural network system. Figure 8 shows the arrangement.

The same figure of merit defined by the norm of the error function can be applied for the selection of the optimum set of values for α and β of the R_p n-dimensional mapping. Figure 9 shows how the norm varies in terms of the parameter values selection and when the R_p 121-dimensional mapping is being used as a neural based discriminator system for events without pileup. In this case, $\alpha = 0.9$ and $\beta = 1.6$ represents the optimal tuning. When pileup was included, $\alpha = 0.8$ and $\beta = 0.7$ became optimal. This significant β reduction for events with pileup is in good agreement with the result described above for the R_p scalar function.

The performance achieved using the R_p 121-dimensional function combined with a neural network was of less than 7 (12.5) % of jets misclassified with 97% electron efficiency, when events without (with) pileup are used. At 99% electron efficiency, less than 8.5 (15) % of the jets are misclassified. Figure 10 shows the output of the neural network for testing events without pileup in the case of feeding its input nodes with the image of the R_p 121-dimensional mapping. Ten nodes in the hidden layer were used.

So far, we have computed the R_p mapping by taking the radius distance referred to the cell of maximum energy deposition. However, the barycenter of the shower can be used instead and the energy deposition can be assumed to be entirely confined to the center of each cell. The coordinates of the barycenter are then calculated as:

$$x_{bary} = \frac{\sum_{i=1}^{121} x_i E_i}{\sum_{i=1}^{121} E_i} \quad (4)$$

$$y_{bary} = \frac{\sum_{i=1}^{121} y_i E_i}{\sum_{i=1}^{121} E_i} \quad (5)$$

where x_i and y_i are the coordinates of cell i in the $\eta \times \Phi$ space. The radius for the R_p

mapping is then defined as

$$r_i = \sqrt{(x_i - x_{bary})^2 + (y_i - y_{bary})^2} \quad (6)$$

The main advantage in adding the extra processing needed for the computation of the barycenter coordinates is to include the energy deposited in the central cell, as it is mapped onto zero by the formula shown in (3) due to its null radius. Therefore, one could expect this alternative formulation of the R_p mapping to be slightly better than the one used before. For 97 (99) % electron efficiency and adding the barycenter processing one obtains 7 (8) % of jets misclassified (without pileup). The optimal parameter values in this case were $\alpha = 0.6$ and $\beta = 1.3$. The small improvement is due to the fact that the neural network is able to identify the events with significantly different barycenters and therefore adding the barycenter information helps little the full discriminator operation.

4.1 RpRing

The application of the R_p n-dimensional mapping produces a weighting profile of the energy distribution in the calorimeter that reveals in a better way the subtle differences in the shower development for electrons and jets. The weighting procedure involves computing the Euclidean distance of each cell to a defined center. As we have seen in Section 3, with rings, the normalization factor there used represented a different weighting procedure, as the energy sequential profile avoids the low-energy contents of the farthest rings to be masked by the higher values present in the innermost rings. Therefore, one could think in merging the two processing techniques and to use the following mapping on the input space of the ring sums (see Figure 11(a)):

$$R_p : X \rightarrow Y \quad (7)$$

$$(E_{r_i}, r d_i) \rightarrow \frac{E_{r_i}^\alpha \cdot r d_i^\beta}{\sum_{i=1}^6 E_{r_i}^\alpha}$$

Here, $r d_i$ represents the radius of each ring measured with respect to the first ring. This method offers the advantage of combining the compactness that the ring structure allows with the high performance obtained using the R_p mapping.

The search of the optimum parameter values for this R_p Ring mapping was performed by using once more the norm of the error function. The values found were $\alpha = 1$ and $\beta = 2.7$. Using this optimum values and feeding the weighted ring sums into a neural network configuration of 6-3-1, the R_p Ring method achieved 97% electron efficiency with less than 9.1% of misclassified jets on events without pileup.

The method can also be redefined by applying first the R_p mapping on the calorimeter cells and then using the image cells for building the rings (Figure 11(b)). This alternative composition works similar to the one described above, but requires more processing. In which results are concerned, both methods perform barely the same.

4.2 RpNet

An integrated system was developed using the R_p multi-dimensional processing and the neural network discriminator. The basic idea is to leave to the network the task of finding the optimum values of α and β parameters, integrating it to the training phase of the network. Figure 12 shows the system. It should be noted that individualized α and β

parameters for each calorimeter cell are now possible, allowing the network to identify by itself the best profile of the weighting function ($E_i^{\alpha_i} r_i^{\beta_i}$).

This concept has the advantage of skipping the search procedure (based on the norm of the error vector) of the optimum set of values for α and β , as the neural processing would perform this search automatically. In the appendix, one can find the mathematical development that supports this method.

The performance achieved with this system was 97% electron efficiency with 7 (11) % error in the identification of jets, when pileup is not (is) included. For 99% electron efficiency, the error increases just to 8.2 (14) %. For this test, β 's have been kept fixed to 2. The final values for α reveal that the network interprets ring sums by means of a weighting profile that boosts the outermost rings. In Table 2, it is shown the average value found by the network for each ring. This result is in agreement with the normalization method used for the ring sums in Section 3, based on the sequential energy profile.

Table 3 resumes the performance of the different R_p methods.

5 Conclusions

The analysis of a second-level triggering system based on simulated calorimetry data was carried out. The system was realized by neural networks and different calorimeter features were tested. A subregion of interest was defined comprising 121 cells of the calorimeter. Sending to the neural network input nodes the total energy deposited in each one of such cells, an electron efficiency of 97% was achieved, while 8.7 (13.8) % of jets were misclassified in case of pileup of 0 (20) minimum bias events. The use of this feature results in good performance and requires few pre-processing (to find the cell of maximum energy deposition).

It was also shown that using concentric rings as input, the performance is slightly worse (10.5 (16.9) % of jets misclassified as electrons for the same electron efficiency) but a faster and more compact network can be designed due to the reduced dimension of the input space. This performance can be improved by either weighting properly the contribution of each cell to the ring sums or by adding the information of variables available from the first-level trigger system.

The best performance was shown to be obtained by applying an extra processing. The combination of the R_p multi-dimensional function and a neural network forms a system that reaches 97% efficiency in triggering on electrons with an error in the identification of jets of 7 (12.5) %. For 99% electron efficiency, the error in jet identification increases to 8.5 (15) %. The optimum value of the R_p parameters can be determined using a method that takes the norm of the error function for jets as a figure of merit. An integrated system based on neural networks, which combines the pre-processing of the input data through a R_p mapping and the discriminating action of the neural network classifier, was also developed. On the other hand, the combination of the discrimination power of the R_p weighting procedure and the compact structure provided by the ring feature improves the ring analysis performance, misidentifying 9.1% of jets when 97% electron efficiency is kept and events without pileup are considered.

The comparison of the performance achieved with the various methods here proposed is illustrated in Figure 13, where we took as a reference the method referred to before [7]. It can be seen that all neural methods proposed perform better than the reference one.

Acknowledgements

We are thankful for the support that has been provided to this work by Conselho Nacional de Desenvolvimento Científico e Tecnológico (Brasília, Brazil) and CERN. We would like to thank the RD11 Collaboration for providing the simulated data sets. In particular, we express our best thanks to R.K. Bock, F. Block, W. Krischer, G. Klyuchnikov, I.C. Legrand and J. Carter, all from CERN, for fruitful discussions concerning this work. We also thank B. Kastrup and H. Gottschalk for helping in parts of the analysis.

Appendix

A RpNet: Mathematical Formulation

Define

$$t_i = \frac{E_i^{\alpha_i}}{\sum_{j=1}^N E_j^{\alpha_j}}, i = 1, N \quad (8)$$

$$p_i = r_i^{\beta_i}, i = 1, N \quad (9)$$

$$u_i = \sum_{j=1}^N W_{i,j} t_j p_j, i = 1, M \quad (10)$$

where N represents the dimension of the input vector (121 in our case), M the number of nodes in the hidden layer (10 in our application) and u_i is the input to the i -th node of the hidden layer.

The updating of α for the backpropagation training method needs $\Delta\alpha$ to be computed, which requires computing the derivative of the instantaneous square error function (ϵ^2) with respect to α_i :

$$\Delta\alpha_i = -\eta_\alpha \frac{\partial \epsilon^2}{\partial \alpha_i} = -2\eta_\alpha \epsilon \frac{\partial \epsilon}{\partial \alpha_i} \quad (11)$$

Δp and ΔW are similarly computed. Using the chain rule, one gets:

$$\frac{\partial \epsilon}{\partial \alpha_i} = \sum_{j=1}^N \frac{\partial \epsilon}{\partial t_j} \frac{\partial t_j}{\partial \alpha_i} \quad (12)$$

$$\frac{\partial t_j}{\partial \alpha_i} \Big|_{j=i} = E_i^{\alpha_i} \ln(E_i) \frac{\sum_{k=1, k \neq i}^N E_k^{\alpha_k}}{(\sum_{k=1}^N E_k^{\alpha_k})^2} \quad (13)$$

$$\frac{\partial t_j}{\partial \alpha_i} \Big|_{j \neq i} = \frac{-E_j^{\alpha_j} E_i^{\alpha_i} \ln(E_i)}{(\sum_{k=1}^N E_k^{\alpha_k})^2} \quad (14)$$

It rests to be computed the derivative of the error function with respect the input variable t_i , which needs the chain rule to be applied once more:

$$\frac{\partial \epsilon}{\partial t_i} = \sum_{j=1}^M \frac{\partial \epsilon}{\partial u_j} \frac{\partial u_j}{\partial t_i} \quad (15)$$

When the gain function is the hyperbolic tangent and only one hidden layer is being used, one gets:

$$\frac{\partial u_j}{\partial t_i} = W_{j,i} p_i \quad (16)$$

$$\frac{\partial \epsilon}{\partial u_i} = [1 - \tanh^2(\mathbf{z}^t \mathbf{x})] z_i [1 - \tanh^2(u_i)] \quad (17)$$

where $\mathbf{z}$ represents the weight vector which connects the hidden layer with the output node and $\mathbf{x}$ is the input of the output node processor.

References

- [1] W.S. Babbage and L.F. Thompson, *NIM A330* (1993) 482.
- [2] J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.
- [3] J. Badier et al. *IEEE Trans. on Nuclear Science* **40** (1993) 45.
- [4] RD-11 Collaboration. Status Report: Embedded Architectures for Second-Level Triggering (EAST). *CERN/DRDC 93-12* (1993).
- [5] H.M.A. Andree et al. Feedforward Neural Networks as an On-Line Pattern Recognition Tool. *Third International Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High-Energy and Nuclear Physics*, pp 315, World Scientific (1994).
- [6] R.K. Bock, J. Carter and I.C. Legrand. What Can Artificial Neural Networks Do For the Global Second Level Trigger. *CERN-ATLAS/DAQ-11* (1994).
- [7] G. Klyuchnikov et al. A Second-level Trigger, Based on Calorimetry Only. *CERN-ATLAS/DAQ-007* (1992).
- [8] T. Sjöstrand. Pythia 5.6 and Jetset 7.3 - Physics and Manual. *CERN-TH/6488-92* (1992).
- [9] RD-27 Collaboration. *CERN-ATLAS/DAQ-005* (1992).
- [10] L. Lönnblad et al. *Comp. Phys. Comm.* **70** (1992) 167.
- [11] J.M. Seixas et al. A Second-Level Trigger System Based on Calorimeters and Using Neural Networks for Feature Extraction and Electron/Jets Discriminations. *Fourth International Conference on Calorimetry for High Energy Physics*, Elba, Italy, 1993.
- [12] D. Acosta et al. *NIM A314* (1992) 431.
- [13] D. Acosta et al. *NIM A305* (1991) 55.
- [14] J.M. Seixas et al. Neural Networks Applied to a Second-Level Trigger System Based on Calorimeters. *Third International Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High-Energy and Nuclear Physics*, pp 353, World Scientific (1994).

Tables

Feature	NN	Performance			
		without pileup		with pileup	
		e^- efficiency	jets error	e^- efficiency	jets error
Squared Rings	6-3-1	97%	10.5%	97%	16.9%
Circular Rings	6-3-1	97%	9.7%	97%	15.9%
Sq. Rings & L1	8-4-1	97%	8.8%	97%	14.5%

Table 1: Results using the rings method.

Ring	$\alpha_{average}$
Ring 1	0.1
Ring 2	0.4
Ring 3	0.5
Ring 4	0.8
Ring 5	0.9
Ring 6	1.0

Table 2: Average values for the α parameter with respect to rings.

Feature	jets error	
	@ 97% e^- efficiency	@ 99% e^- efficiency
R_p 1-Dim	9%	11%
R_p N-Dim	7%	8.5%
R_p Ring	9.1%	14%
R_p Net	7%	8.2%

Table 3: Performance of R_p methods for events without pileup.

Figure Captions

1. Typical electron (a) and jet (b) events without pileup. E.m. and hadronic sections were added in a cell to cell basis.
2. Neural network output for the matrix method. An 11x11 matrix is used.
3. Building squared (a) and circular (b) rings. Numbers in (b) show the fraction of energy from each cell that contributes to the respective ring sum. In (a) the weight is unitary for all cells.
4. Defining $E_{cluster}$ and E_{isol} variables.
5. The target and ideal curves used to compute the norm of the error function.
6. The norm of the error function for the one dimensional R_p as function of α and β parameters, for events without (a) and with (b) pileup.
7. Electron and jet distributions for the R_p function with $\alpha = 0.9$ and $\beta = 3.6$. Only events without pileup are considered.
8. Neural network fed with energy values of each cell after multidimensional R_p pre-processing.
9. The norm of the error function for the neural system using the multidimensional R_p pre-processing as a function of α and β parameters.
10. Neural network output for the multidimensional R_p method.
11. The R_p Ring method. In (a) the R_p maps the ring sums and in (b), the R_p maps the cell values and rings are built from the mapped cells.
12. Integrating the multidimensional R_p pre-processing to the neural network discriminator: the RpNet method.
13. Comparison of the methods.

Figure 1

Figure 2

(a)

(b)

Figure 3

Block of calorimeter

$\Delta n \times \Delta \phi \approx 0.02 \times 0.02$

Figure 4

Figure 5

(a)

(b)

Figure 6

Figure 7

Figure 8

Figure 9

Figure 10

(a)

(b)

Figure 11

Figure 12

Figure 13