UNIVERSITE DE MONS-HAINAUT

Physique Générale et Physique des Particules Elémentaires

X

Neural network technique for point source search using the '97 AMANDA data: Comparison with sequential cut method

PPEl-UMH 2003 03363

Thierry Castermans
Fernand Grard

March 3, 2003

Neural network technique for point source search using the '97 AMANDA data: Comparison with sequential cut method

PPEl-UMH 2003 03363

Thierry Castermans
Fernand Grard

March 3, 2003

1. Introduction:

The problem of the rejection of background and selection of signal in a sample of data events can in principle be solved by using the neural network and in particular the multi-layer perceptron technique as available in PAW [See write-up of J. Schwindling and B. Mansoulié (Saclay) and O. Couet (CERN)]

The main purpose of this report is to show what can be achieved using the neural network technique by comparison with the results obtained by one of us (TC[1]) using an optimized sequential method ("pedestrian method") for the rejection of background and selection of signal from the AMANDA '97 data in an attempt of finding evidence for astronomical point sources.

Applying appropriate cuts on the following analysis variables:

- Zenith(2)
- Ndirb(2)
- Ndirc(6)
- Ldirb(6)
- Smootallrl(6)
- Ndira(2)
- Smootallphit
- Zenith(6)
- (cogz: not included because of 100% selection efficiency for signal and background)

the following end result has been obtained by TC:

- $\varepsilon_s = 9$ %
- $\varepsilon_b = 0.005$ %
- R = 1650

2. Exploitation of the PAW neural network:

The configuration defining parameters of perceptron (N_{in} - N_{hid} - N_{out}) are:

- N_{in} = number of input neurons = number of analysis variables considered
- N_{hid} = number of hidden neurons: variable
- N_{out} = number of output neurons: 1

The choice of analysis variables is restricted to those which have been used for the sequential cut method.

The range of variation of the different analysis variables has been normalized to (0 , 1)

[1] Thierry Castermans : « Recherche de sources ponctuelles de neutrinos d'origine astrophysique avec le détecteur AMANDA » Université de Liège, 2002.

The error function E which has to be minimized in order to find the best values of the neuronic weights insuring efficient separation between background and signal is constructed as follows:

$$E = \frac{1}{2}\sum_{p} w_p \left(o_p - t_p\right)^2$$

where p runs over all the events contained in the training samples, o_p is the output value for event p, w_p is a weight which may take into account the relative population of the training samples or which may be attached to each individual event p. The parameter t_p is the expected output value (0 if the event p is background; 1 if signal).

There are 7 learning methods available as they have been enumerated by the above authors:

1 Stochastic minimization (often wrongly called "standard online back propagation")
2 Steepest descent with fixed steps ("batch back propagation")
3 Steepest descent with line search
4 Conjugate gradients with Polak-Ribiere updating formula
5 Conjugate gradients with Fletcher-Reeves updating formula
6 Broyden-Flectcher-Goldfarb-Shanno (BFGS) method
7 Hybrid linear- BFGS method

- The training samples consist of Monte-Carlo background events and atmospheric neutrino signal events.

The number of learning steps is 1000.

Guide lines: According to the above mentioned authors:

- "In principle, one hidden layer is sufficient …"
- "There are no rules to choose the number of hidden neurons… empirical way… by following the evolution of the error with the numbers of *learning steps*…"
- "*To avoid overfitting the learning samples should be large enough*… There seems to be no strict rule on the ratio Number of *events*/Number of weights, which should be between 10 and 100"
- "It is better to start with a small number of neurons: learning is faster, often enough, avoids overfitting problems"
- Performances should be evaluated from samples of events different from the training samples

- **Evaluation of the performances of the neural network relative to number of hidden neurons and to learning method.**

For this purpose, the same analysis variables as for the sequential cut method are used (8 variables).

For convenience, the training is performed using the events of the Monte-Carlo sample not taking into account the individual weights

The performances are determined in terms of:

- signal efficiencies (ε_s)
- background efficiencies (ε_b)

- ratio R = $\varepsilon_s / \varepsilon_b$
- error function E at the end of training

for different number of hidden neurons and for the different available learning methods.

Target values for the efficiencies are either $\varepsilon_s \approx 9\%$ or $\varepsilon_b \approx 0.01\%$.
Perhaps in a more objective way, the cut on the neural output value has in some cases been defined in such a way that the product:

$$\Pi = R \times (\varepsilon_s - \varepsilon_b)$$

is maximum, so requiring simultaneously the largest R and the largest difference of efficiencies. The combination $(\varepsilon_s - \varepsilon_b)$ as adopted by TC as an objective selection criteria happened to have a too broad distribution to be used for the selection of the output events.

The training of the neural network has been made using equally populated samples of background and signal events (4000 each):

$$N_b = N_s$$

1) **Performances as a function of learning method**:

Perceptron 8 – 10 – 1

The number of neuronic weights is equal to (8x10+10) = 90. The ratio " Number of events/Number of weights" is: 8000/90 = 89, which is between the limits as recommended above.

For the evaluation of relative performances, the cut on the output value has been set such that the efficiencies are comparable to what has been obtained by the pedestrian method. The maximum value of Π will appear in the last column of the table when used for the selection (otherwise between brackets for information)

Learning method	ε_s (%)	ε_b (%)	R	Error fct	Π
1	4.9 7.9	0.01 0.024	370 325	0.34	(23.2) 25.7
2					
3	3.6	0.01	355	0.36	
4	**7.2** **3.4**	**0.013** **0.001**	**551** **3011**	**0.32**	**(39.7)** **102**
5	0.7	0.01	70	0.41	
6	14.0 2.0	0.2 0.12	61 17	0.31	
7	0.05 9.8	0.07 0.14	0.7 71	0.31	

According to our criteria, learning methods 1 and 4 happen to be the best with a preference for method 4.

Fig.1(a, b, c) shows from learning method 4 :

- The distribution of events as a function of neural output value for signal, background and data

- The evolution of the error function as a function of number of learning steps
- The efficiencies ε_i, the efficiency ratio $R = \frac{\varepsilon_s}{\varepsilon_b}$ and the difference $(\varepsilon_s - \varepsilon_b)$ for signal, background and data selection above cut on the neural output value

2) <u>Performances as a function of number of hidden neurons:</u>

The following tests have been made with learning method 1.

<u>Learning method 1</u>

N_{hidden}	ε_s (%)	ε_b (%)	R	Error fct.
5	4	0.01	370	0.34
10	**4.9**	**0.01**	**469**	0.33
	0.4	0.00019	2380	
20	**6.2**	**0.014**	**455**	0.33
	5.1	0.0077	670	
	3.5	0.0033	1082	
50	**6.5**	**0.01**	**704**	0.33
80	**6.2**	**0.013**	**488**	0.33

The performances are not very sensitive to the number of hidden neurons. Even with 80 hidden neurons, the ratio: "Number of events/Number of weights" is still between the recommended limits.

3) <u>Performances as a function of the ratio $\frac{w_s}{w_b}$ for the training of the neural network</u>.

As usual, equally populated samples of background and signal events have been used (4000-4000)

<u>Perceptron 8 – 10 – 1</u>

$\frac{w_s}{w_b}$	Learn. meth.	ε_s (%)	ε_b (%)	R	Error fct	Π
1	1	4.9	0.01	370	0.34	(23.2)
		7.9	0.024	325		25.7
	4	3.4	0.001	3011		102.0
		7.2	**0.013**	**551**		**(39.7)**
		8.9	0.026	344		X
1.5	**4**	**4.3**	**0.007**	**626**		
2	**4**	**9.0**	**0.013**	**674**		**Max**
		3.5	0.0008	-		X
	1	6	0.01	572		X
		4.5	0.005	824		Max:
3	**4**	**8**	**0.011**	**704**		**X**
		5.8	0.004	1556		Max:
5	4	10.4	0.15	69		Max:
10	1	0.6	0.01	59	0.67	

Remarks:

- It is surprising to observe that the neural network output range differs generally from the expected (0, 1) range.
- Learning method 4 happens to be the best considering the various test conditions.

- **Comparison of performances from the sequential cut technique and the neural network technique**

For this purpose, the individual weights of the Monte Carlo events in the signal and background samples are taken into account.

Here also, target values for the efficiencies are either $\varepsilon_s \approx 9\%$ or $\varepsilon_b \approx 0.01\%$ and in some cases the maximum of the product Π has been considered.

- Learning methods 1 and 4 are used.

- The number of events in the training samples has been chosen according to the relative average individual weights so as to get about equal influence from the background and from the signal in the training procedure.
  - Average signal weight = 0.071
  - Average background weight = 0.115
  - Therefore, training samples containing 6480 signal events and 4000 background events have been used (same ratio as 0.115/0.071 = 1.62)
  - Training with increased number of signal events has also been tried

- Perceptrons with different number of hidden neurons have been tested

Efficiencies are evaluated from the whole samples available of MC background and signal events, individual weights being taken into account.

- Background: 1.885.823 entries → 216.343 effective events
- Signal: 441.855 entries → 31.558 effective events

Efficiencies of data selection are also given from a data sample amounting to 1.588.052 events.

The error function as displayed on the figures have been evaluated:

- upper curve: with the test samples
- lower curve: with the training samples

No cut on Zenith(2) or on Zenith(6) before training

N_s	N_b	Config.	Learn meth.	ε_s (%)	ε_b (%)	ε_d (%)	R_s	$\varepsilon_d/\varepsilon_b$	ErrFct	Π.
6480	4000	8-10-1	4	9.8 12.1 **13.6** 14.7	0.002 0.005 **0.009** 0.013	0.047 0.088 **0.12** 0.15	4368 2305 **1517** 1109	23.5 16.6 **13.3** 11.5		428
6480	4000	8-10-1	1	**7.2** 9.1 12.1	**0.013** 0.02 0.05		**536** 387 230			(38.6) (34.9) (27.8)
6480	4000	8-20-1	4	3.1 **7.2** 9.1 12.1	0.001 **0.013** 0.02 0.05		3987 **536** 387 230		0.117	(123.9) (38.6) (34.9) (27.8)

6480 x1.5	4000	8-10-1	4	**10.1** 11.9	**0.010** 0.019		**1045** 615	**0.121**		
6480 x1.5	4000	8-20-1	4	3.9 7.9	0.002 0.017		2324 464	0.123		
6480 x1.5	4000	8-50-1	4	**10.5** 11.7	**0.010** 0.013		**1090** 855			
6480 x1.5	4000	8-80-1	4							
6480 x2	4000	8-10-1	4	14.8	0.067		222			32.7
7029 → 6480	5219 → 4000	8-10-1	4	15.0	0.055		270			40.3

Again, training method 4 looks better and the results are not very dependent on the number of hidden neurons.

Fig. 2 shows the distributions of the selected data events as a function of Zenith(6):

(a) as obtained by the sequential cut technique of TC

(b) as resulting from perceptron 8 – 10 – 1, learning method 4, no cut, N_s/N_b = 6480/4000

It can be seen that the distributions are quite different depending on the method being used: the events being preferentially selected near the horizon with the sequential cut method, or toward the nadir with the neural network technique.

Changing the conditions of exploitation of the perceptrons did not lead to significant changes in the sky distributions.

Fig. 3(a, b, c, d) shows the Zenith(6) distributions from the MC signal and background training samples and the corresponding distributions after selection by the perceptron.

As a result of this observation, the neural network performances were investigated with special consideration to the two analysis variables Zenith(2) and Zenith(6).

The following guiding principles for the training procedure have then been followed:

- Only one Zenith angle is introduced in the neural network: Zenith(2) and Zenith(6) are strongly correlated. This would reduce redundancy information fed to the neural network.
- Zenith(6) will be fed to the neural network: this variable, which is the result of iterative maximum likelihood fitting, is a priori more reliable than Zenith(2).
- No rejection cut will be applied to Zenith(2): events with Zenith(2) < 90° may appear as events with Zenith(6) > 90° and would otherwise be lost. Their proportions

$$\frac{N\left[\left(\theta_{Z,2} < 90°\right) \rightarrow \left(\theta_{Z,6} > 90°\right)\right]}{N\left(\theta_{Z,2} < 90°\right)}$$

  happen to amount to 4.3% for background and to 3.6% for signal.
- Since only events with Zenith(6) > 90° are kept for the final analysis after neural

network selection, only events with Zenith(6) > 90° will be used for the training: the consideration of events with Zenith(6) < 90° is useless and would just contribute to dilute the useful information fed to the neural network.

By doing so the selection efficiencies for signal and for background are not exactly comparable to those obtained above but we still adopted the same target values.

N_s	N_b	Config.	Lmet	ε_s (%)	ε_b (%)	R_s
5968	4000	7-10-1	4	12.5	0.011	1092
				11.9	**0.010**	**1226**
				10.6	0.006	1762

No significant difference with respect to separation of background and signal events as well as to final Zenith(6) distributions is observed.

Fig. 4 shows the resulting Zenith(6) distribution of the selected data events.

By comparing the Zenith(6) distributions (signal, background, data) before and after the selection of events, one observe that either method (sequential cut or neural network) favours either the horizon region or the nadir region. We were then led to apply the neural network technique separately on events from three different regions of the sky as shown in the following table:

Sky region	N_s	N_b	Config.	Lmet	ε_s (%)	ε_b (%)	R_s
90<Z(6)<120	5375	4000	7-10-1	4	23.6	0.357	67
120< Z(6)<150	**7094**	**4000**	**7-10-1**	**4**	**11.8**	**0.018**	**671**
150< Z(6)<180	5455	4000	7-10-1	4	66.2	6.4	10.3

The number of events in the training samples have been determined according to the same rules as above.

It should be remarked that the separation of events works in a rather acceptable way in the central region only.

Similar results were obtained by TC by application of the sequential cut method on the same three regions of the sky.

Conclusions:

As far as overall selection efficiencies are concerned.

- Learning method 4 appeared to be the best, although learning method 1 is acceptable.
- The performances of the perceptrons are not very sensitive to the number of hidden neurons, at least when the ratio: "number of training events/number of internal weights" are within the recommended limits.
- The relative populations of the training samples should be adjusted so that the training is equally influenced by background and by signal events, although small deviations are tolerated.

The resulting strong distortion of the Zenith(6) distributions is to be attributed to the choice of the set of analysis variables and to their degree of correlation with the Zenith(6) parameter as well as to the particular shape of their distributions. This situation should be investigated further to better determine the neural network performances with respect to the present problem of separating signal and background in an Amanda-like experiment.

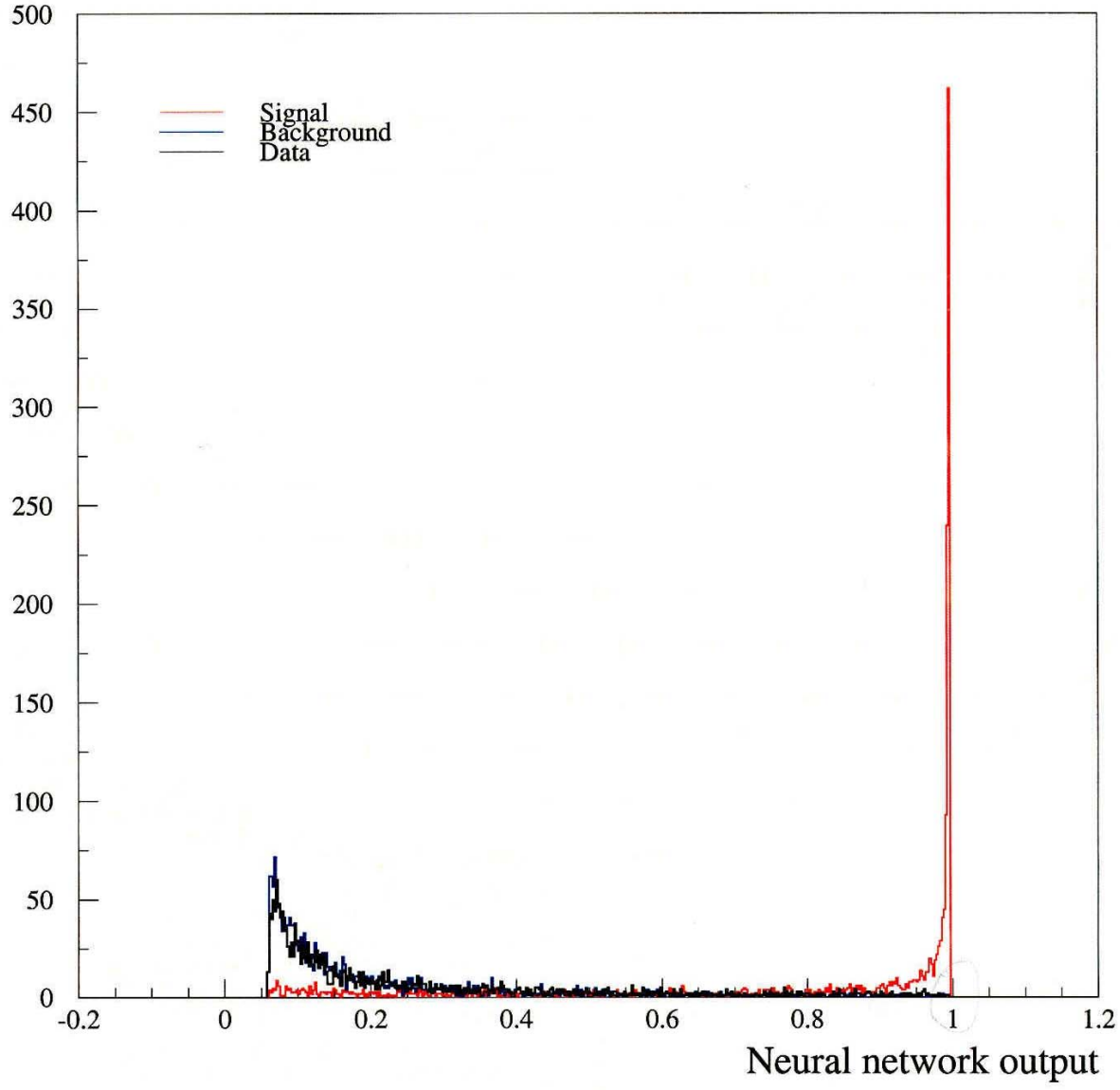

Fig. 1a : Distribution of the events as a function of the neural network output value for signal MC, background MC and experimental data.

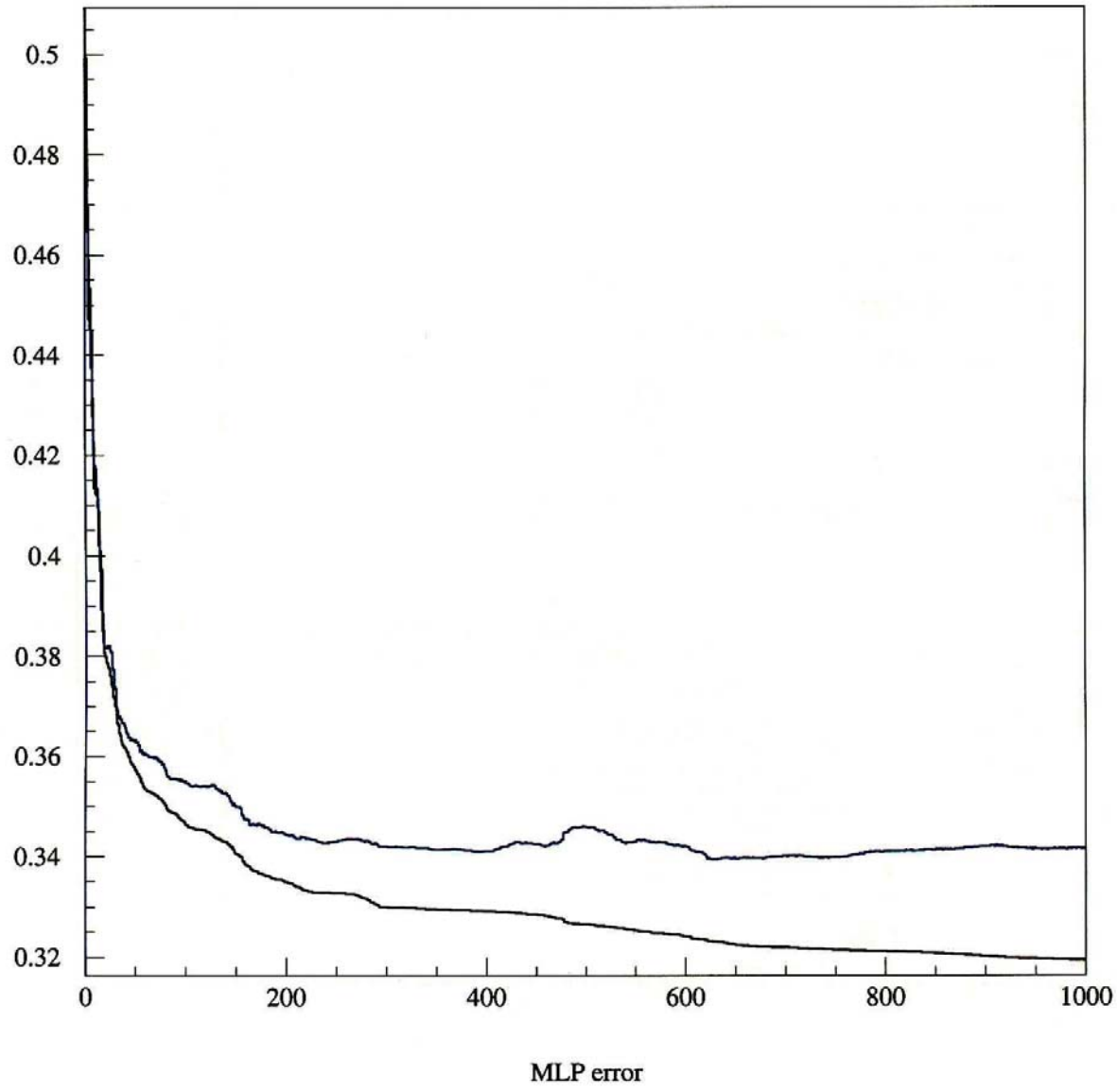

Fig. 1b : Evolution of the error function as a function of the number of learning steps. The upward (downward) curve corresponds to the test (training) sample.

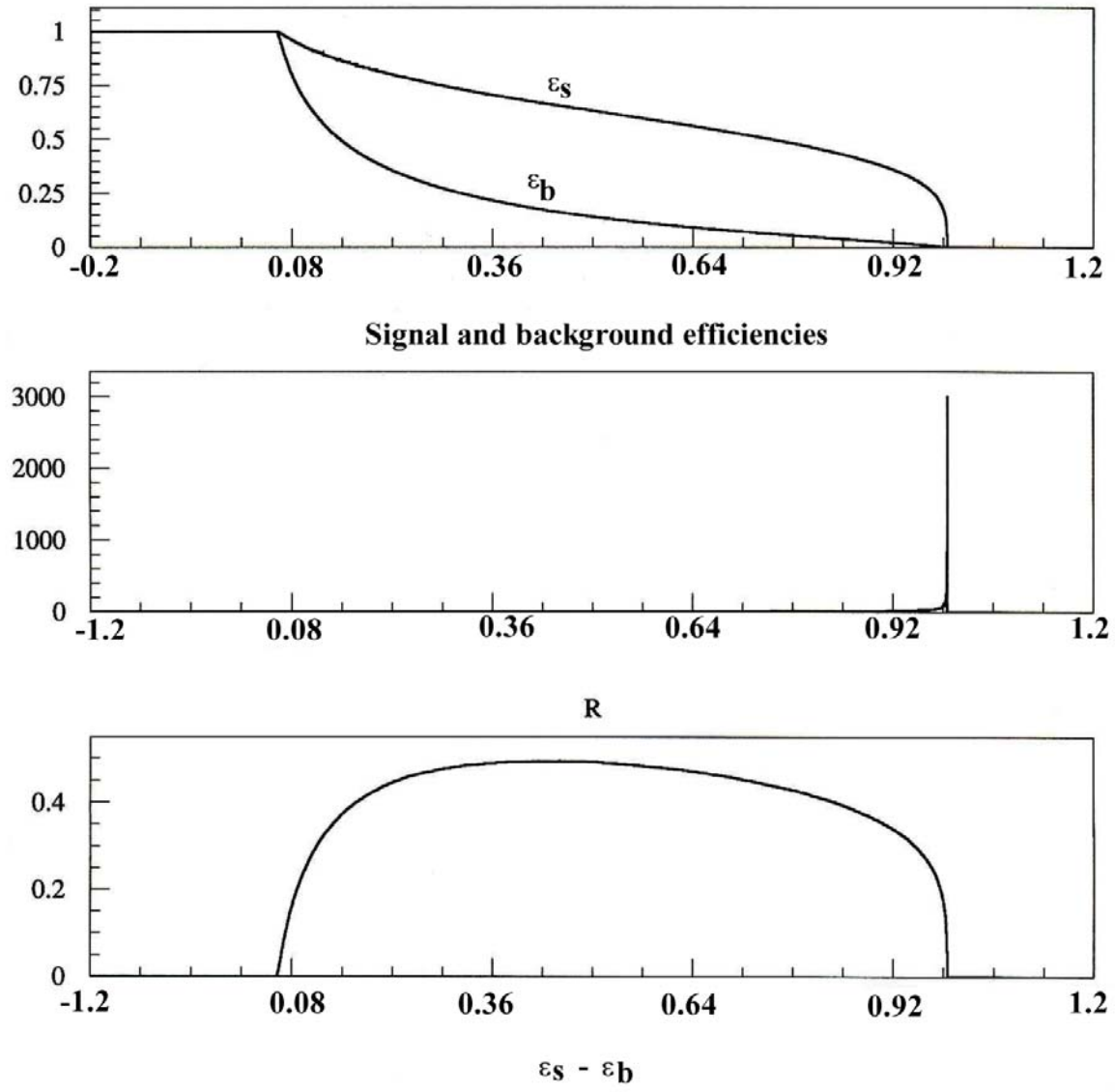

Fig. 1c : Efficiencies, efficiency ratio $R = \varepsilon_s/\varepsilon_b$ and difference $(\varepsilon_s - \varepsilon_b)$ for signal, background and data selection above cut on the neural network output value.

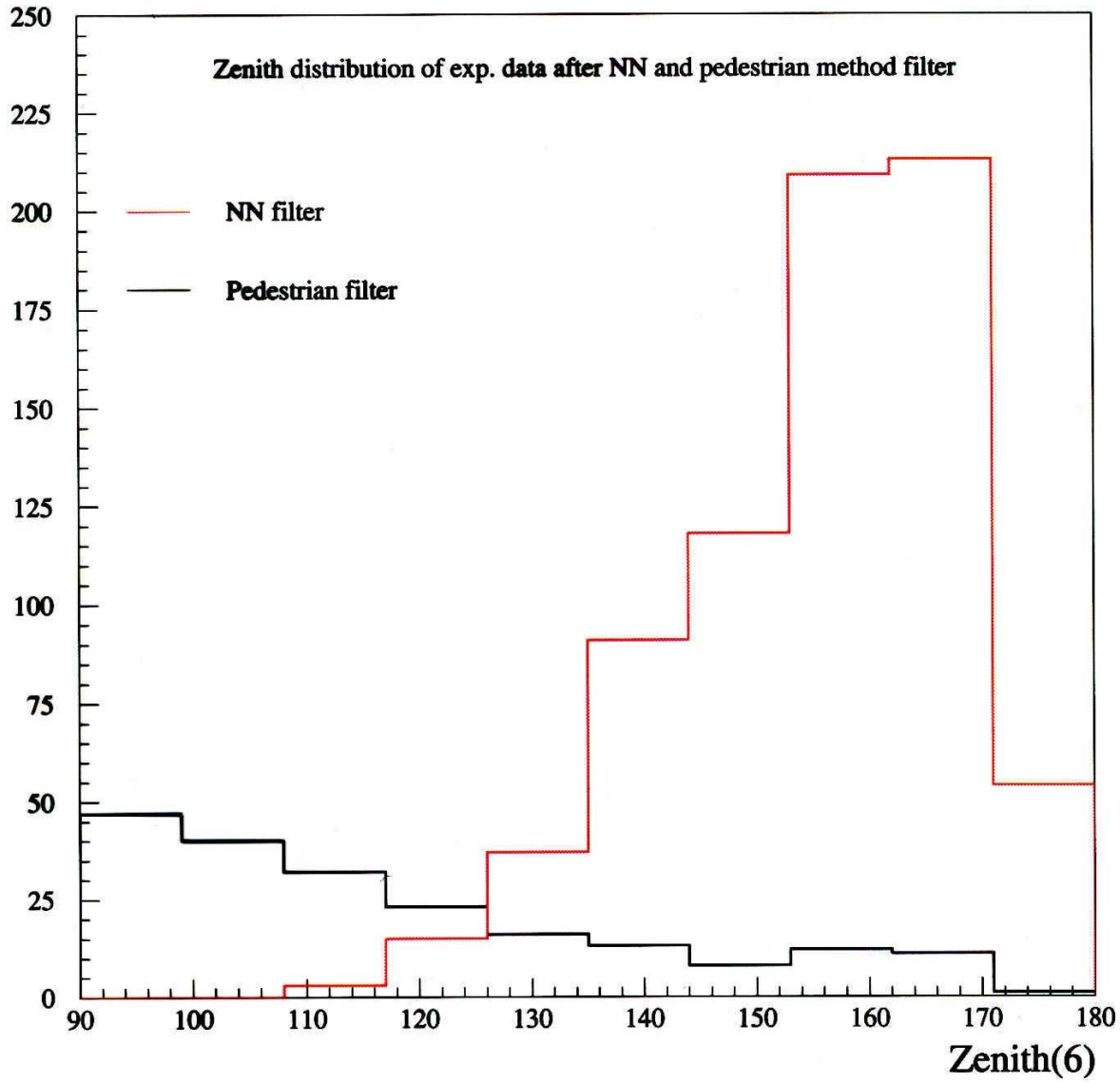

Fig. 2 : Distributions of the selected data events as a function of zenith angle (16 iterated Pandel likelihood fit) as obtained by the sequential cut technique of TC (black curve) and as resulting from the perceptron 8-10-1 using learning method 4 (red curve).

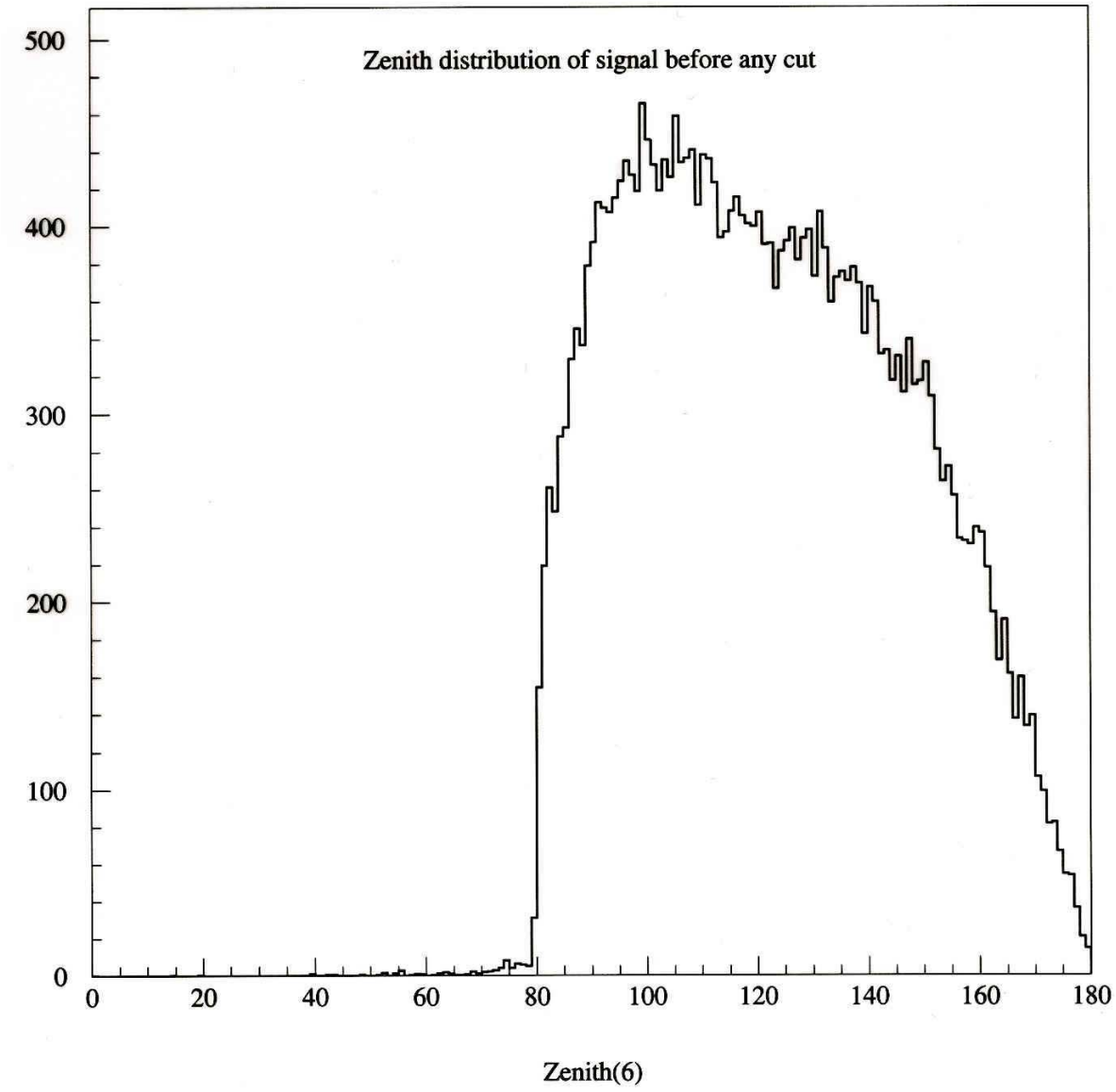

Fig. 3a : Zenith distribution of signal MC before any cut.

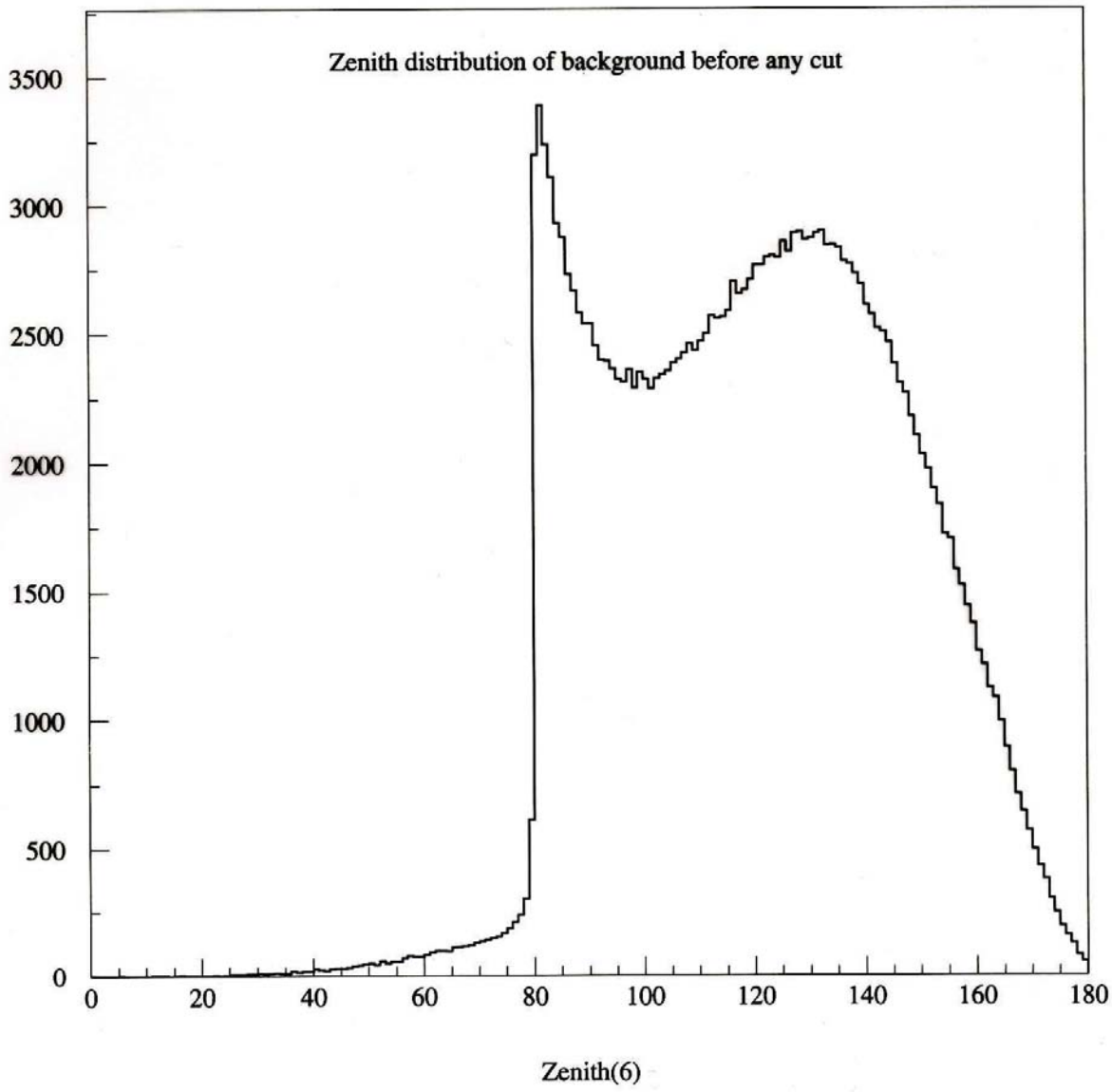

Fig. 3b : Zenith distribution of background MC before any cut.

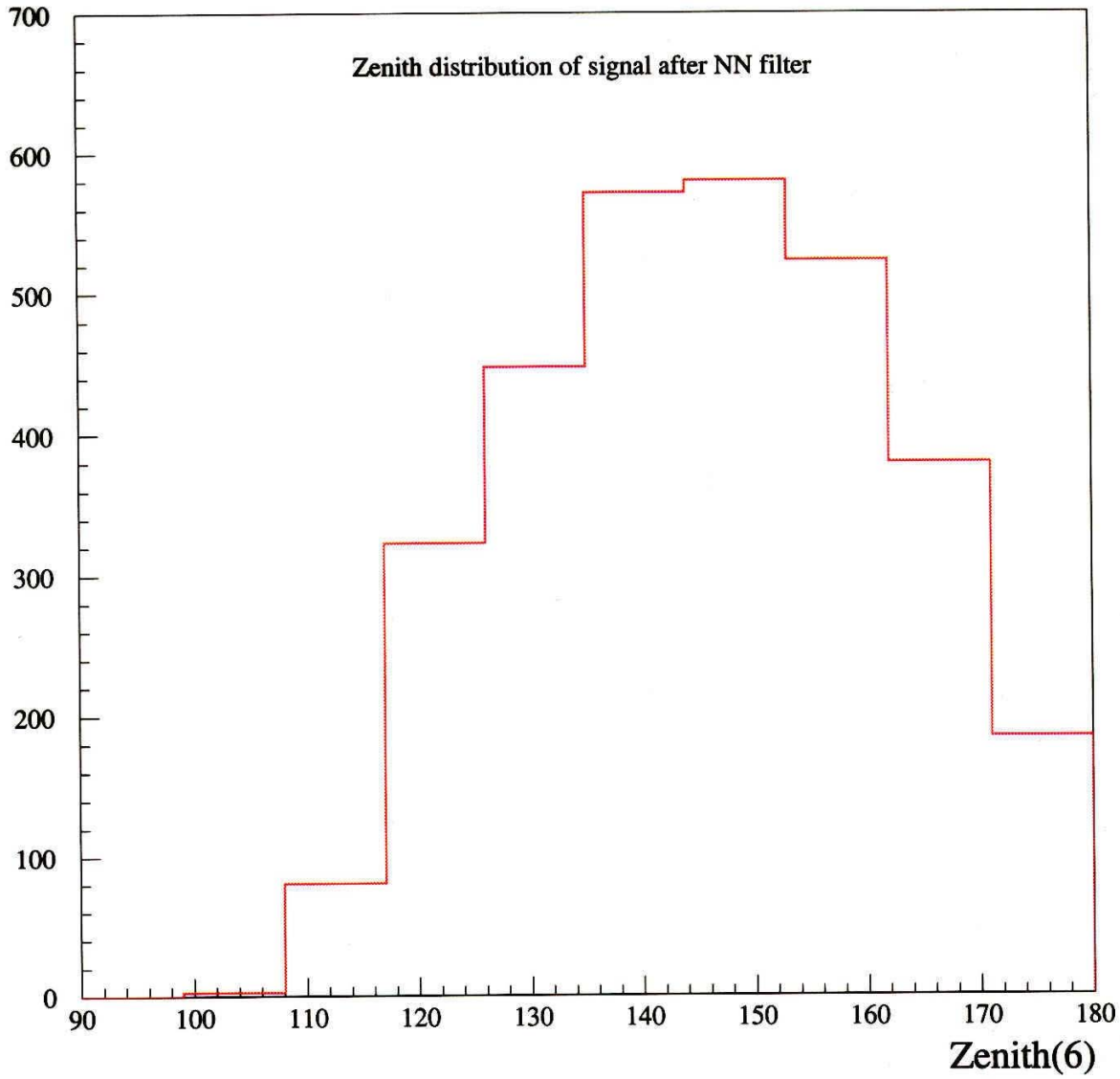

Fig. 3c : Zenith distribution of signal MC after selection by the neural network.

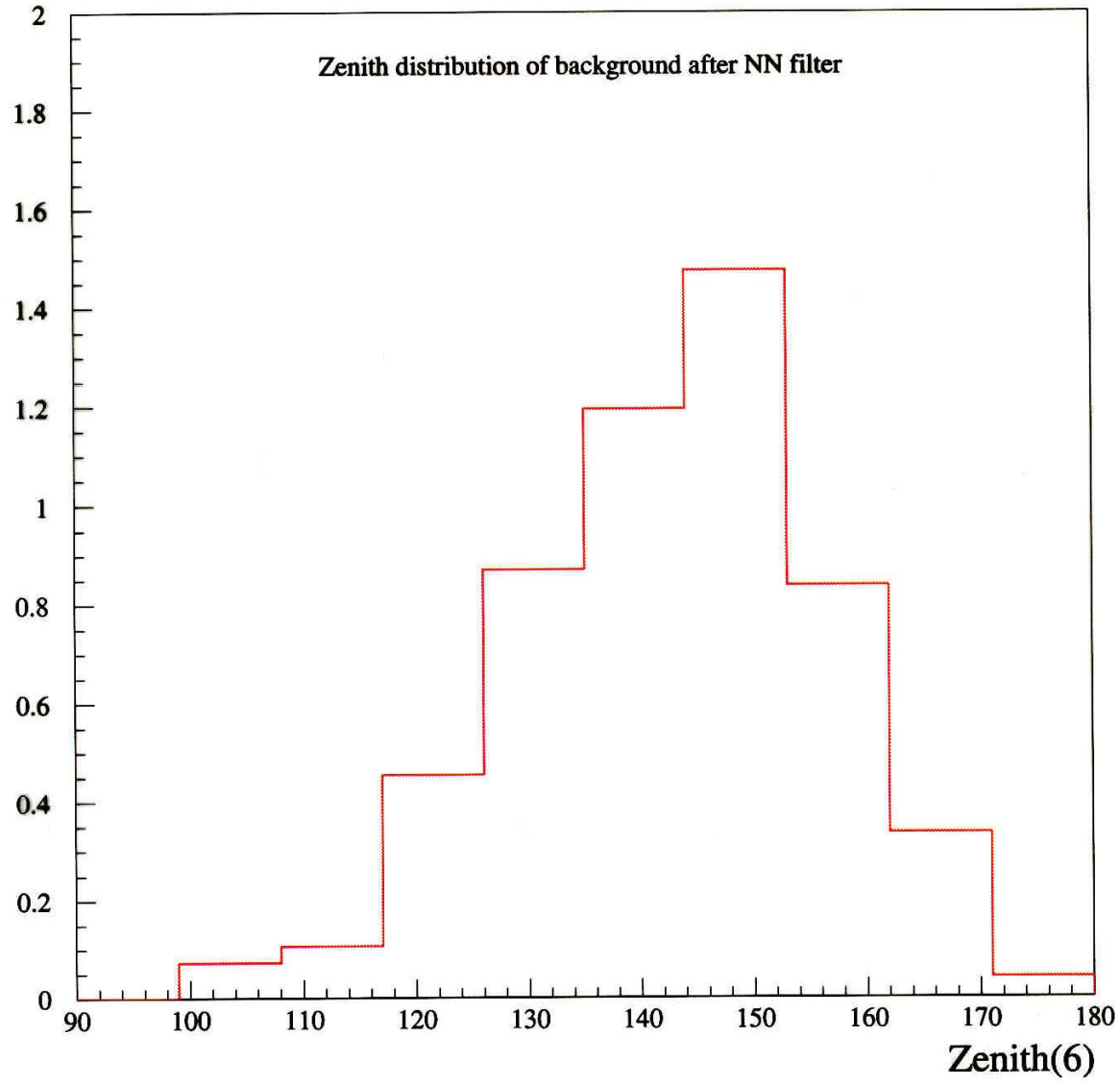

Fig. 3d : Zenith distribution of background MC after selection by the neural network.

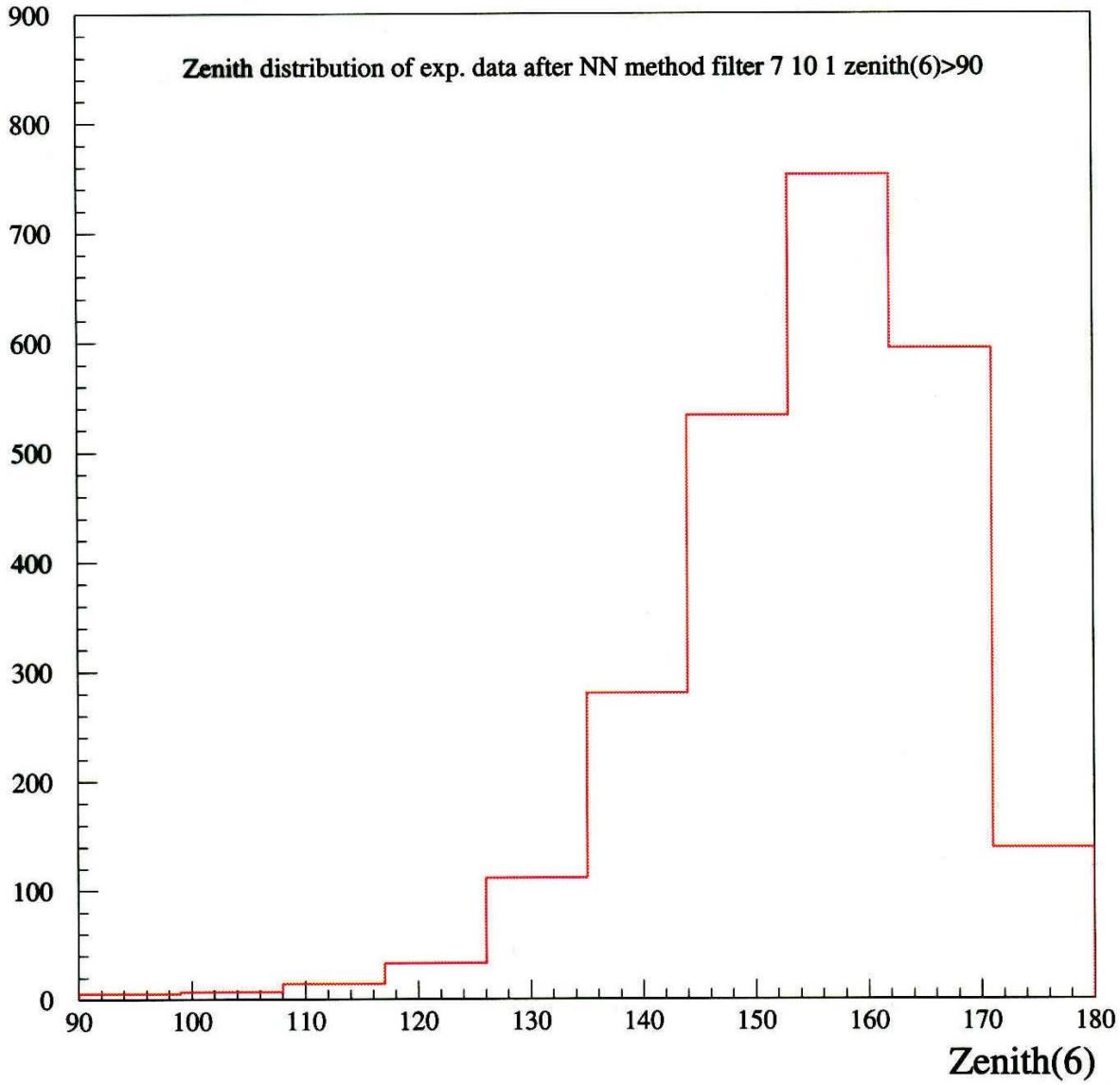

Fig. 4 : Zenith distribution of experimental data after selection by the neural network 7-10-1 and the cut zenith(6) >90.