Regularization methods

vs

large training sets

J. J. Vega[1], *H. Carrillo Calvet[2] and J. L. Jiménez Andrade[2]*

[1]Departamento del Acelerador, Gerencia de Ciencias Ambientales, Instituto Nacional de Investigaciones Nucleares

[2]Laboratorio de Dinámica no Lineal, Facultad de Ciencias, Universidad Nacional Autónoma de México

jaime.vega@inin.gob.mx

OUTLINE

Introduction

Bragg Curve Spectroscopy (BCS)

Digital pulse shape analysis of Bragg curves using ANN

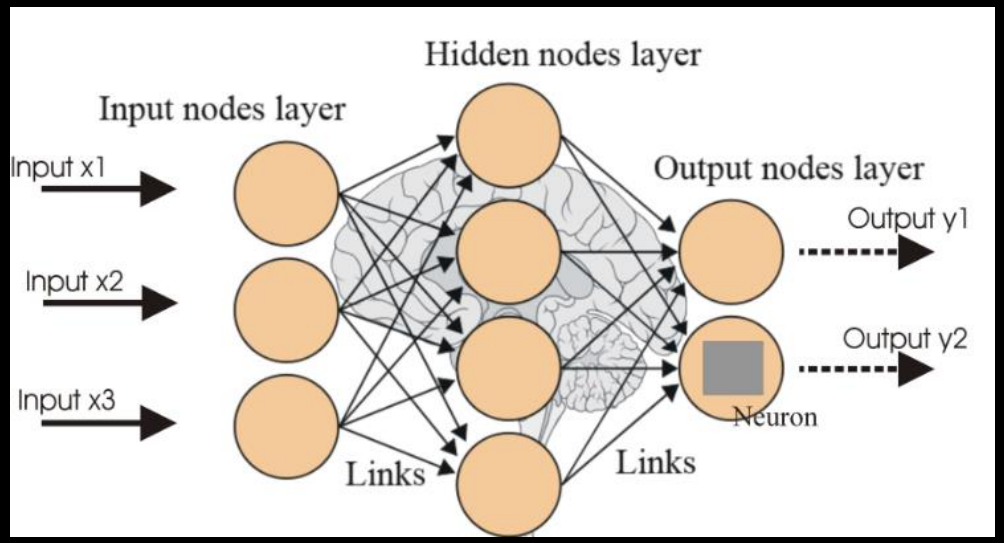
Intercomparison of 3 ANN training alternatives

Results

Discussion

Conclusions

INTRODUCTION



Artificial neural network (ANN)

Digital pulse shape analysis (DPSA)

Data acquisition system

ANN are adaptive systems that show advantages for DPSA.

Two related ever-present problems in the diverse applications of ANNs as pattern recognizers (signal pulses) are overfitting and overtraining. In this work we will focus ourselves on the analysis of three common methods that people use to address these two problems. This will be accomplished in relation to the problem of identifying the signals (pulses) coming from a Bragg curve spectrometer.

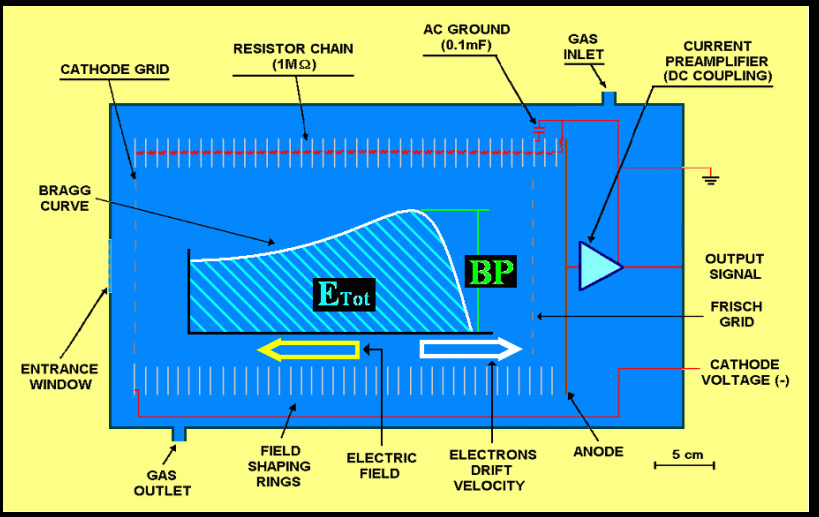
Overfitting relates to the complexity of the model used to fit the training dataset. It means using an ANN with more parameters justifiable by the complexity of the dataset. The network may memorize the data in the training set, including noise, and, consequently, will generalize poorly to new situations, so one might expect that there is an optimum number of parameters that gives the best generalization performance, corresponding to the optimum balance between overfitting and underfitting.

Overtraining has to do with the presence of signal noise capable of overshadowing the smaller and subtle pattern real features. Once an ANN learns all the pattern features not hidden by the concomitant noise, overtraining will manifest itself if training is further continued, because rather than learning the noise hidden features, it will start learning the noise component present in the training patterns, with the consequent worsening of the ANN generalization capability.

The analysis of overfitting and overtraining problems have been considered widely in the literature [Mo-92, Te-95, Sa-95, Bi-95, Ha-96 and Bi-06]. Here, we will present an intercomparison of the advantages and disadvantages of **three alternatives that may be used to reduce overtraining and overfitting effects** when training an ANN.

BRAGG CURVE SPECTROSCOPY (BCS)

Stopping power = S(E) = dE/dx ≡ Bragg curve



AMP → BP^0

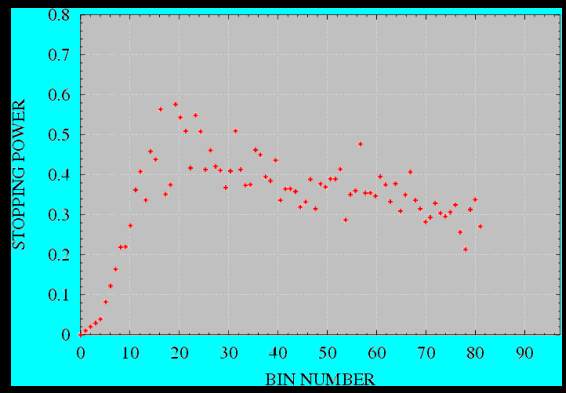
Small integration time, BP signal

AMP → E^0_{Tot}

Large integration time, E_{Tot} signal

Two output parameters: Bragg Peak and Total Energy

Multiparametric output (Bragg curve)

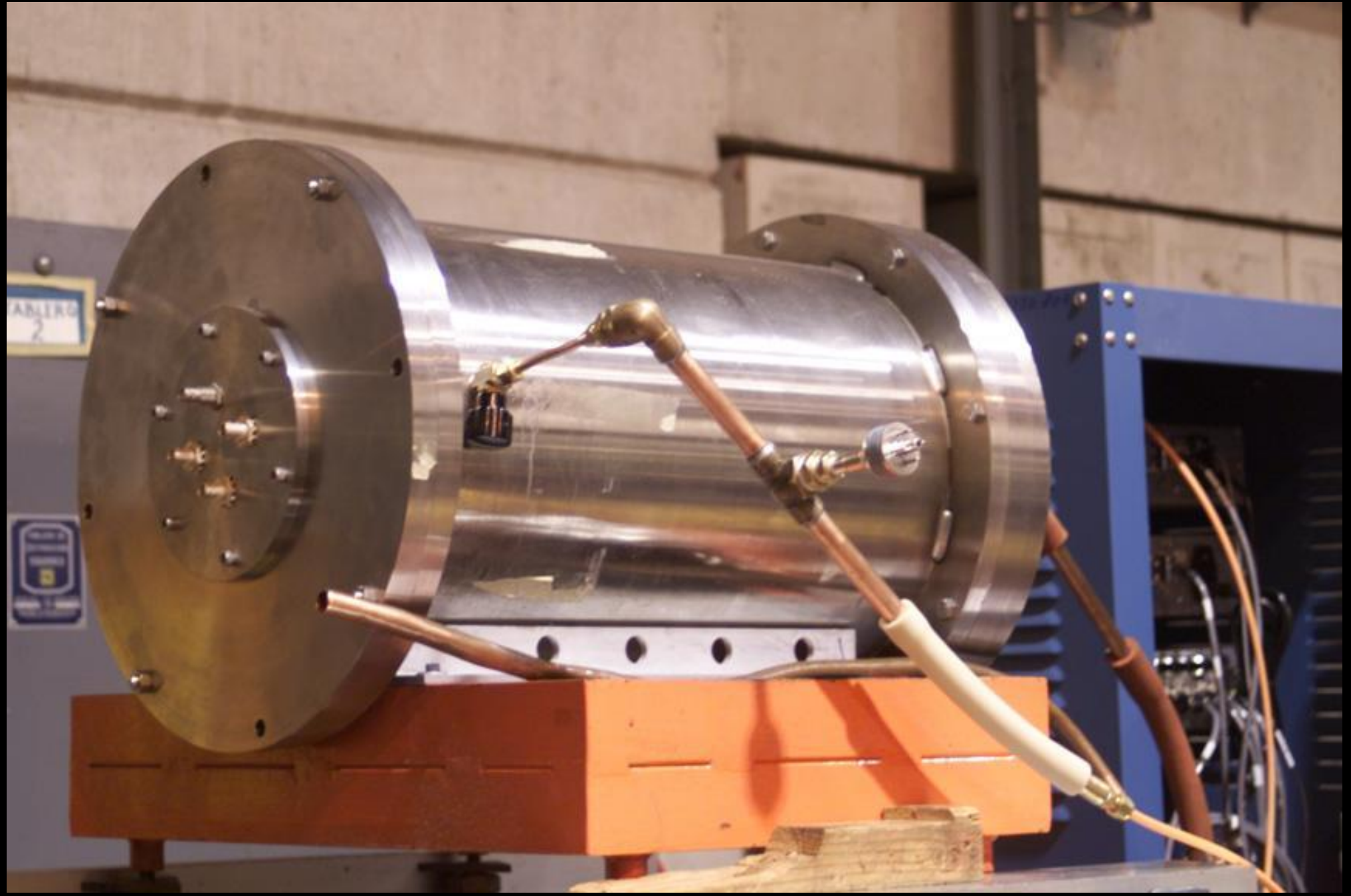


Input from BCS

Waveform digitizer

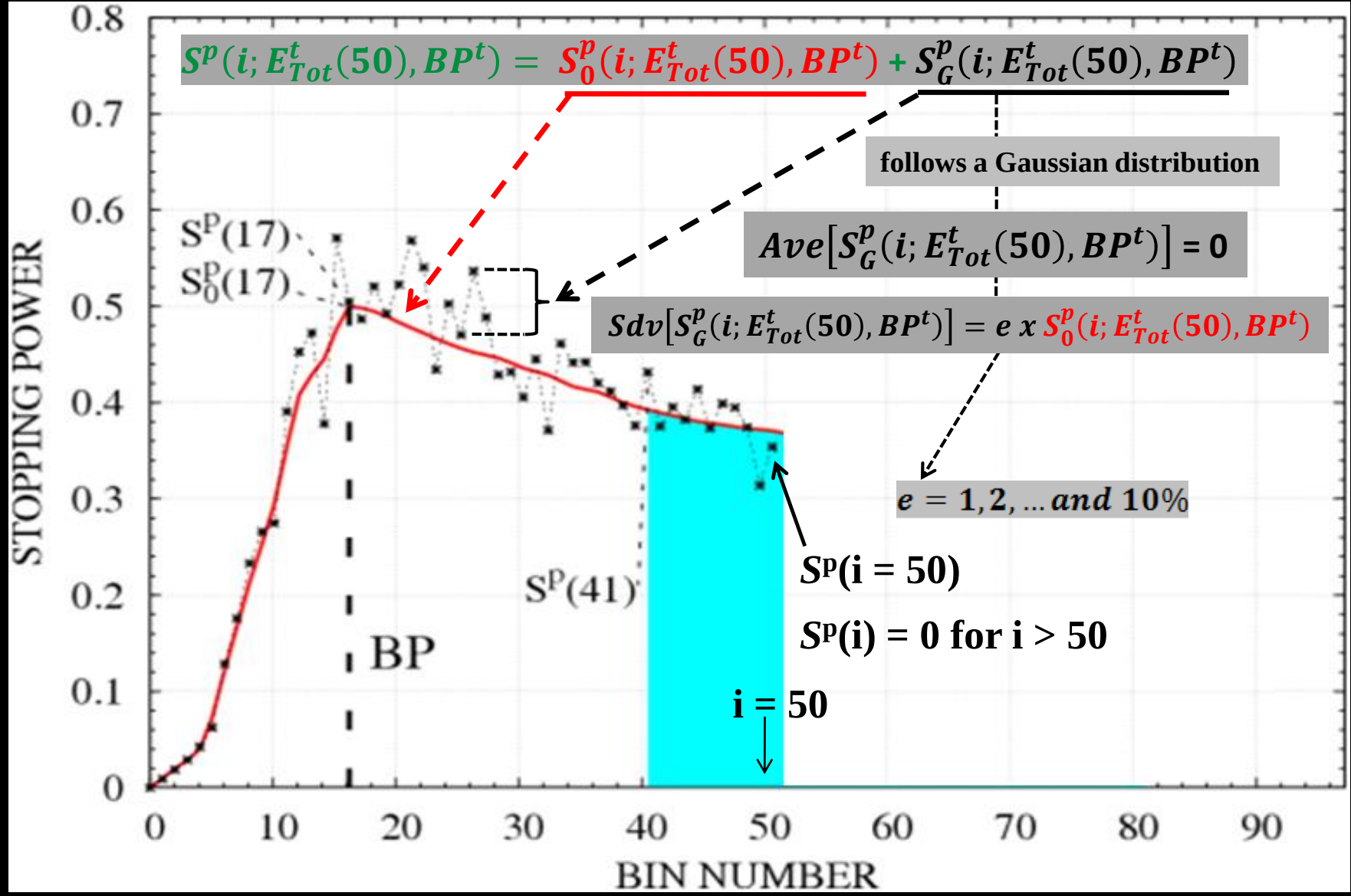
The multiparametric option requires: a powerful data acquisition system, the development of data analyzing and processing software based in new computational paradigmas.

Ref. [Gr-82, Mo-84 and Vi-87]



Bragg curve spectrometer

DIGITAL PULSE SHAPE ANALYSIS OF BRAGG CURVES USING ANN



We will employ synthetic Bragg curves (BCs) in order to test the Digital Pulse Shape Analysis (DPSA) used in [Ve-06] to do BC identification. These BCs will be saved as arrays of a maximum length of 81 values, $\{S(i)\}_{i=0,80}$, as shown in the figure.

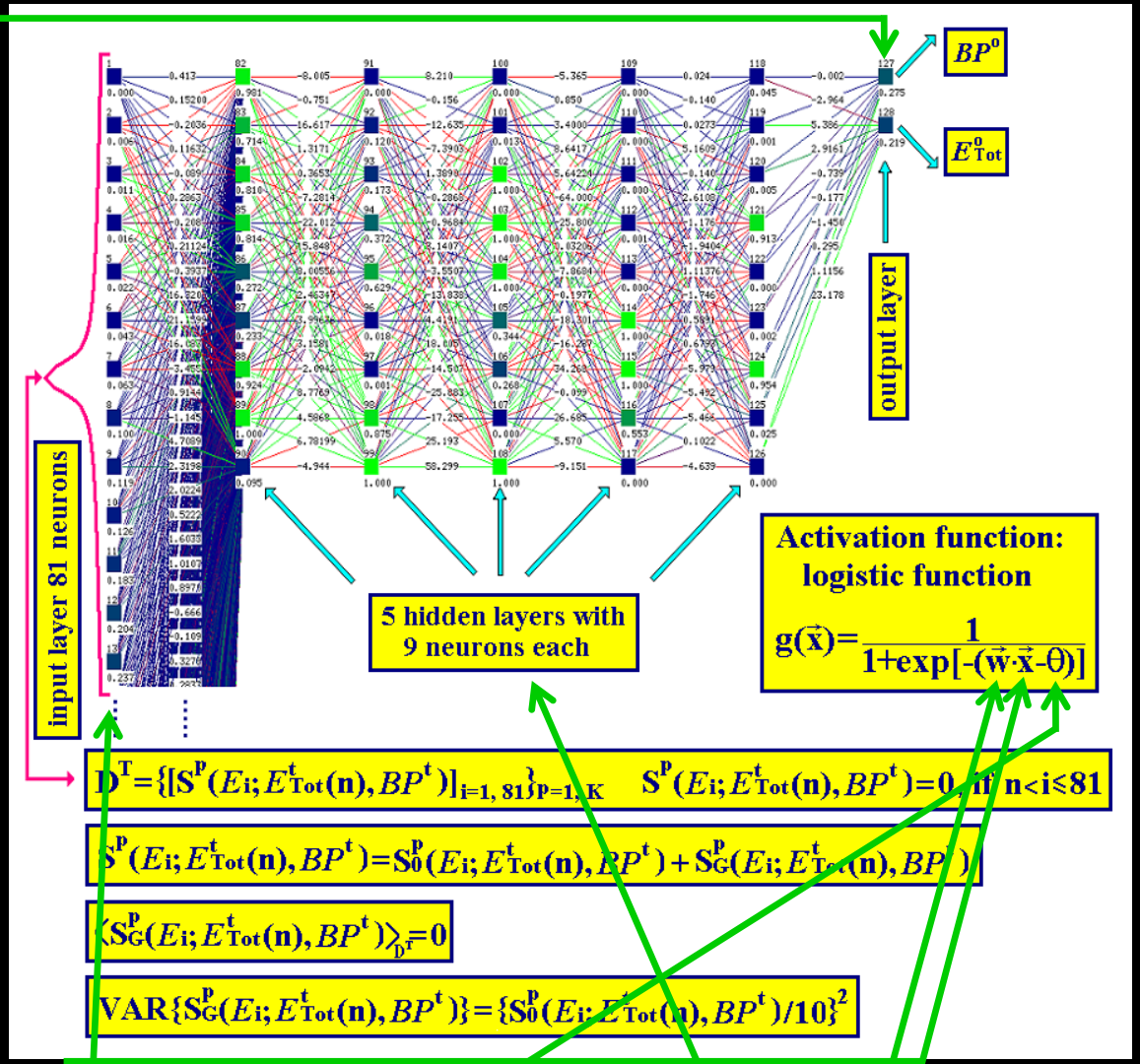
To appreciate how an ANN learns its assigned task, it is convenient to look at the evolution of the corresponding sum of square error curves.

$$E^T(\rho) = \frac{1}{K}\sum_{p\in D^T}\left|\vec{y}^t - f\left[\vec{x}_p;\ \vec{\omega}(\rho)\right]\right|^2$$

and

$$E^V(\rho) = \frac{1}{K}\sum_{p\in D^V}\left|\vec{y}^t - f\left[\vec{x}_p;\ \vec{\omega}(\rho)\right]\right|^2,$$

where $\vec{\omega}(\rho)$ represents the ANN synaptic weight array after ρ training epochs, K is the number of patterns in the sets D^{T} and D^{V}, $f\left[\vec{x}_p;\ \vec{\omega}(\rho)\right]$ is the ANN output for pattern p after ρ training epochs and $\vec{y}^t$ is the target value.



The used ANN architecture: input layer (81 neurons), 5 hidden layers (9 neurons each) and an output layer of 2 neurons. $\vec{x}$ is the neuron input array with information coming from all neurons from the previous layer and pondered by the weight array, $\vec{\omega}$, associated with the links that carry each one of the inputs, and θ is the neuron bias value.

INTERCOMPARISON OF 3 ANN TRAINING ALTERNATIVES

We trained three ANN using different procedures. In the three cases we used a backpropagation learning algorithm.

- **1) Learning rate η = 0.3, momentum term α = 0.15, and early stopping as regularization method. The training, D^T, and validation, D^V, datasets consist of 45,100 BCs.**

- **2) Learning rate η = 0.3, momentum term α = 0.15, and early stopping as regularization method. The training, D^T, and validation, D^V, datasets consist of 451,000 BCs.**

- **3) Learning rate η = 0.3, and weight decay as regularization method. The training, D^T, and validation, D^V, datasets consist of 45,100 BCs.**

Learning law

In the first and second cases, we included a momentum term in the learning law:

learning rate → η, momentum term → μ

$$\Delta\omega_{ij}(\rho+1) = \eta\delta_j o_i + \mu\Delta\omega_{ij}(\rho)$$

In the third case, we included a weight decay term in the learning law:

learning rate → η, weight decay term → λ

$$\Delta\omega_{ij}(\rho+1) = \eta\delta_j o_i - \lambda\omega_{ij}(\rho)$$

RESULTS

An important issue when using weight decay as a regularization method has to do with the way used to reduce the weight decay term, λ.

Weight decay protocol

As the ANN training progresses, it gets complex, that means the absolute value of the synaptic weight array needs to grow, consequently one has to diminish the value of λ, allowing an effective control of the growth rate of $|\vec{\omega}(\rho)|$. The initial λ value we used is 2.5×10^{-9}.

start checking $\Delta\|\vec{\omega}(\rho)\|$	every (epochs)	common ratio	# of steps from 2.5×10^{-9} to 1.0×10^{-9}
2,000	1,000	0.984034	57
20,000	10,000	0.984034	57
200,000	100,000	0.984034	57
2,500,000	100,000	0.998392	569

In figs. 1a-c we present the sum of square error curves $E^T(\rho)$ y $E^V(\rho)$ for each one of the three studied cases.

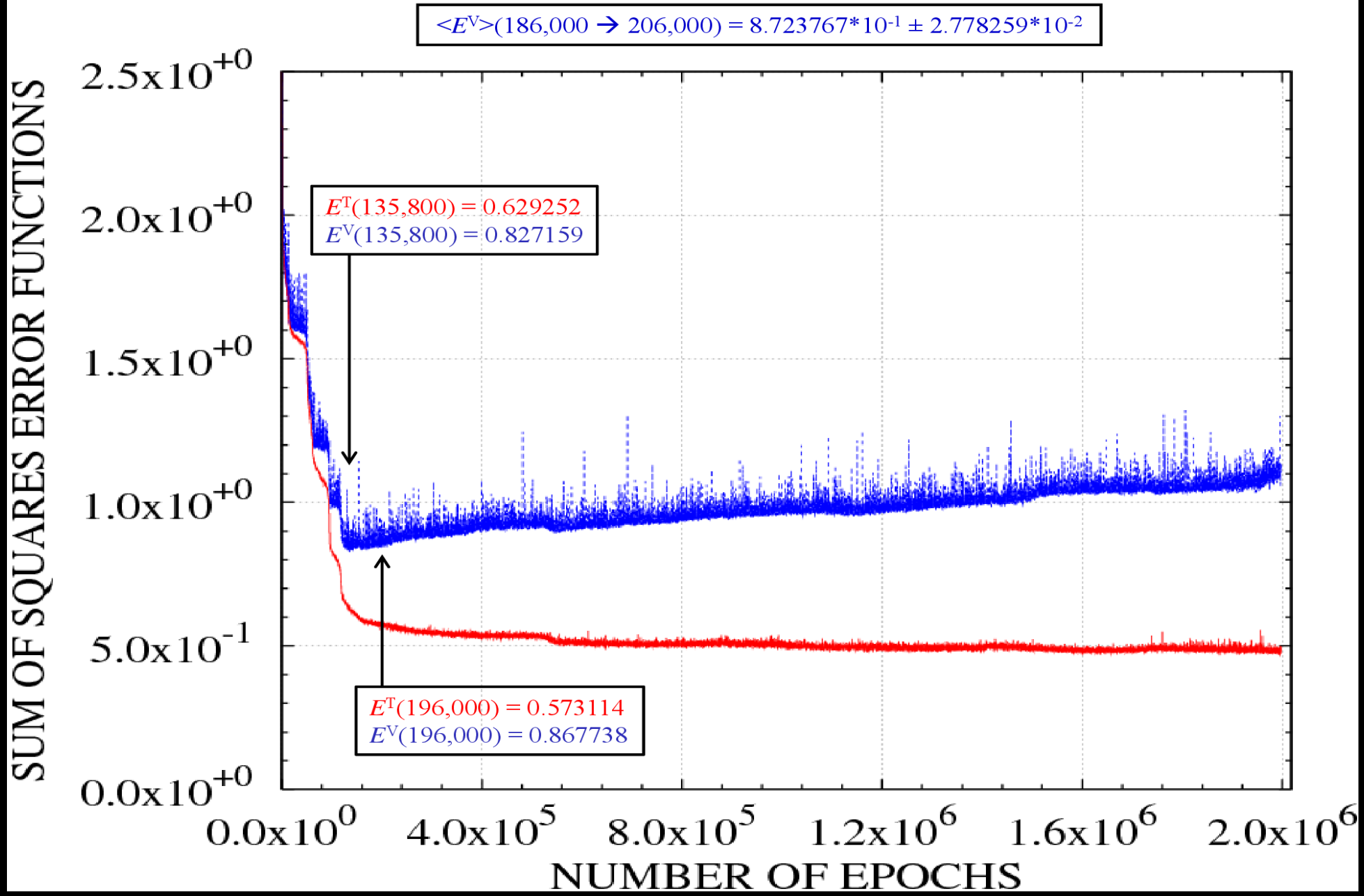


Fig. 1a. Error curves $E^T(\rho)$ and $E^V(\rho)$ for a backpropagation learning law for $\eta = 0.3$, $\alpha = 0.15$ and 45,100 patterns in D^T and D^V.

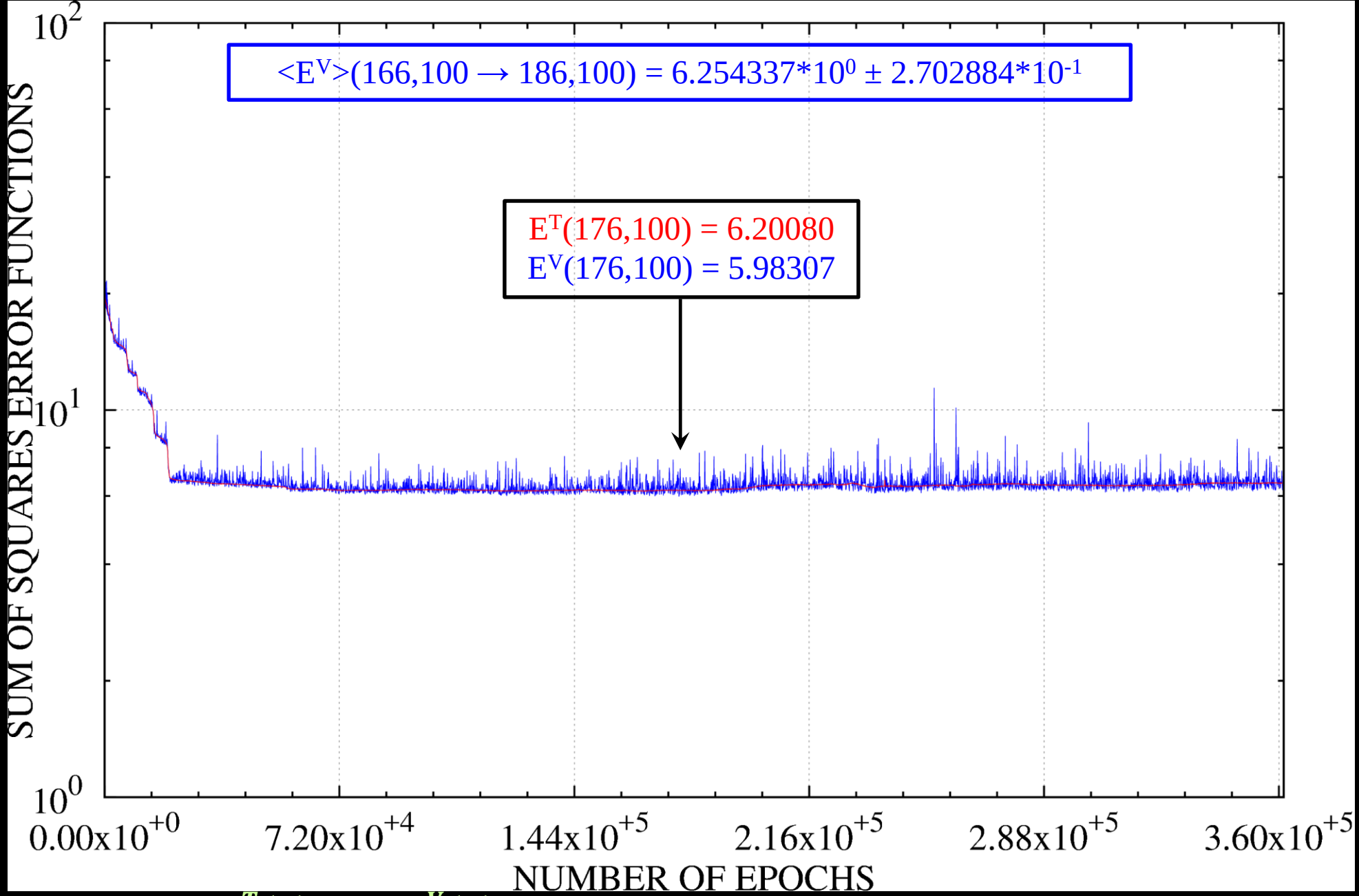


Fig. 1b. Error curves $E^T(\rho)$ and $E^V(\rho)$ for a backpropagation learning law for η = 0.3, α = 0.15 and 451,000 patterns in D^T and D^V.

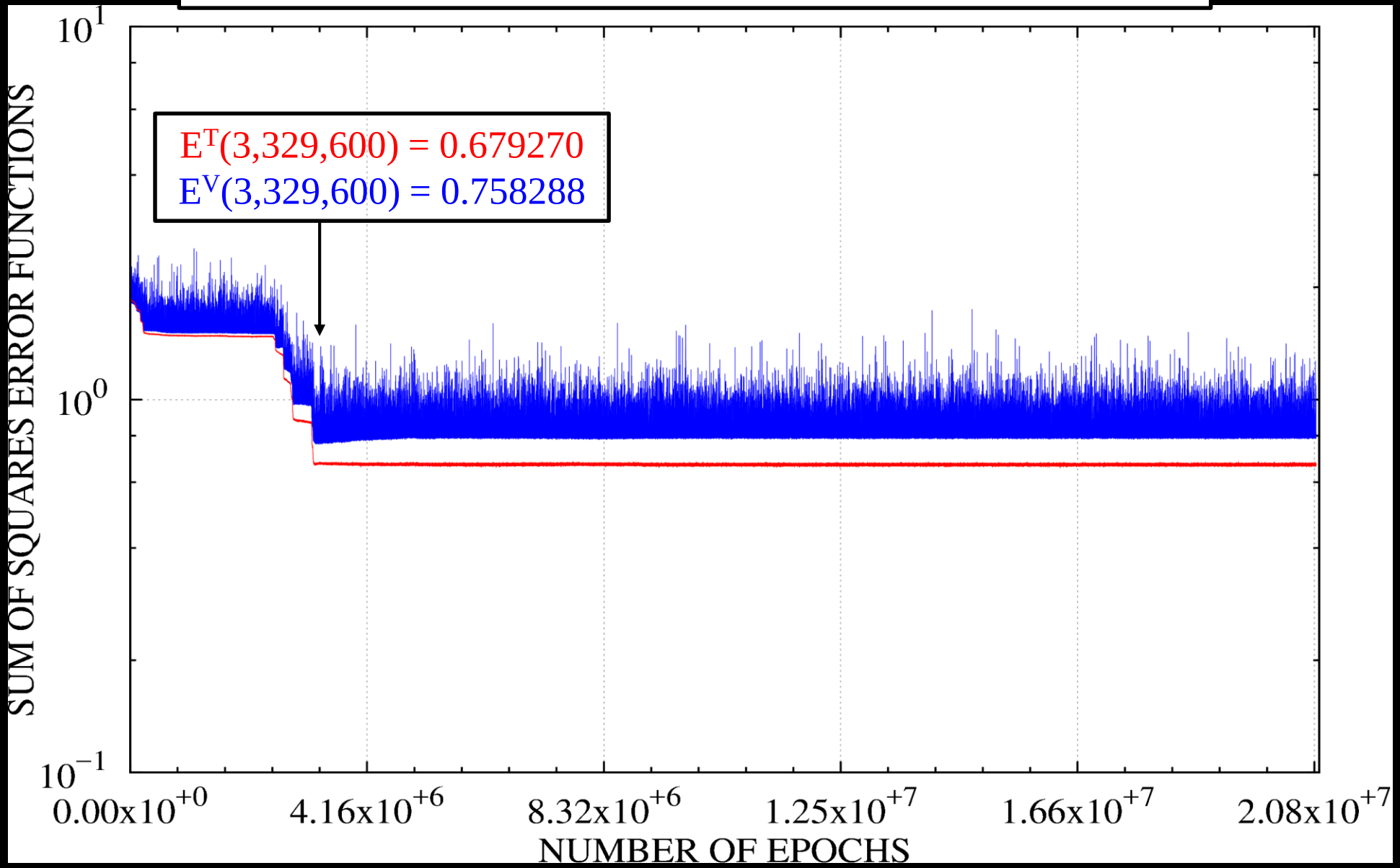


Fig. 1c. Error curves $E^T(\rho)$ and $E^V(\rho)$ for a backpropagation learning law for η = 0.3, a weight decay term with an initial value λ = 2.5x10^{-9}, and 45,100 patterns in D^T and D^V.

Table 1. Average minimum values of the $E^V(\rho)$ curves for each one of the three studied cases, together with the intervals we used to evaluate the three $E^V(\rho)$ average values.

1	$\langle E^V \rangle$(	186,000 →	206,000)	= $8.723767 \times 10^{-1} \pm 2.778259 \times 10^{-2}$
2	$\langle E^V \rangle$(	166,100 →	186,100)	= $6.254337 \times 10^{0} \pm 2.702884 \times 10^{-1}$
3	$\langle E^V \rangle$(	3,319,600 →	3,339,600)	= $7.877160 \times 10^{-1} \pm 2.483593 \times 10^{-2}$

← rescale down!

In Figs. 2a-c, we present the scatter plots corresponding to Figs. 1a-c. Fig. 2d is similar to Fig. 2b, but now, to be able to make a fair comparison of the scatter plots of the three cases, we display only 100 point per class, as in Figs. 2a an 2c. It will be seen that the best scatter plot corresponds do the second case. We obtained these scatter plots for the number of epochs where each one of the $E^V(\rho)$ curves reaches its minimum value.

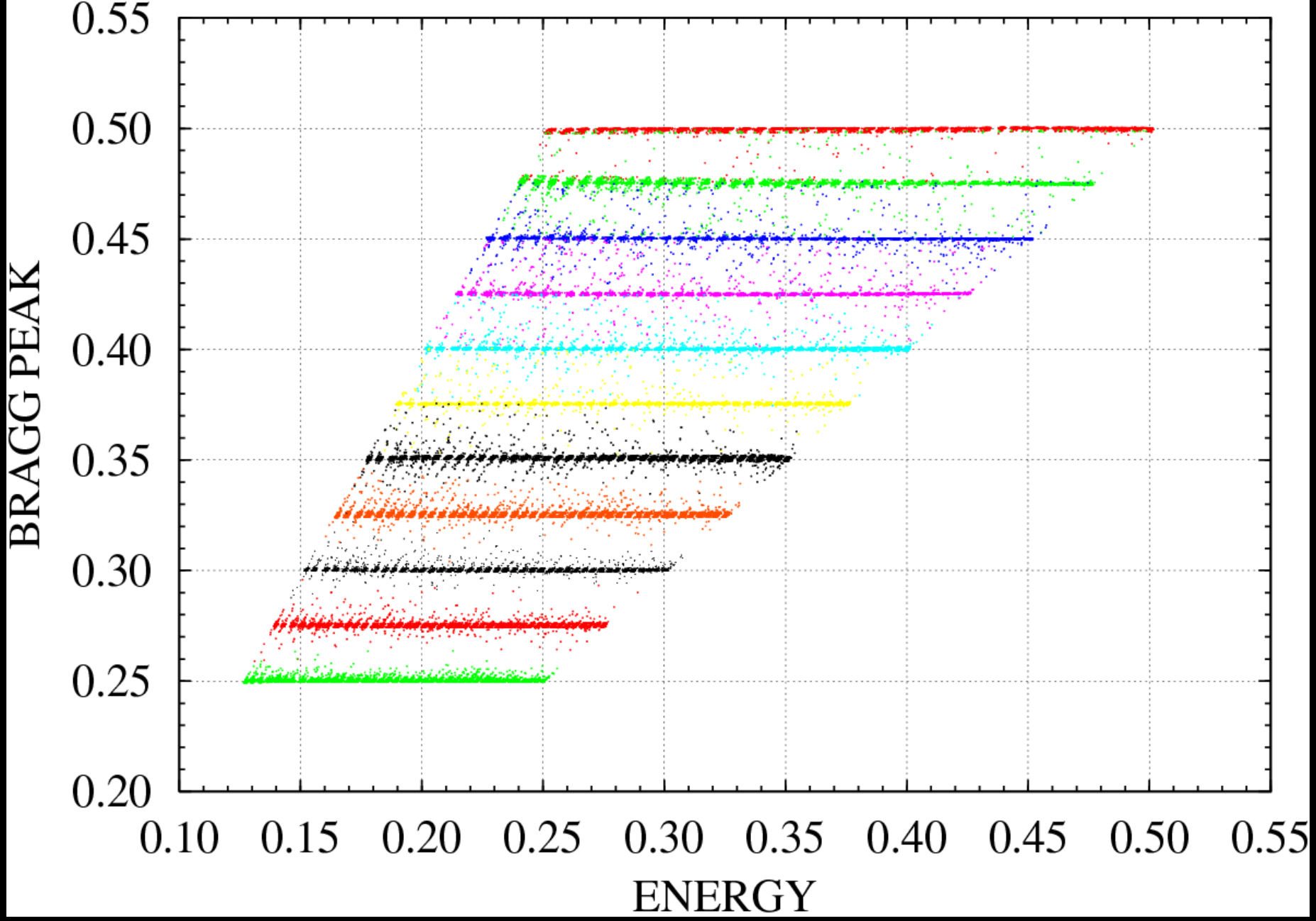


Fig. 2a. Scatter plot for a backpropagation learning algorithm using η = 0.3, α = 0.15 and a datasets with 45,100 BCs. This scatter plot is obtained after 196,000 training epochs.

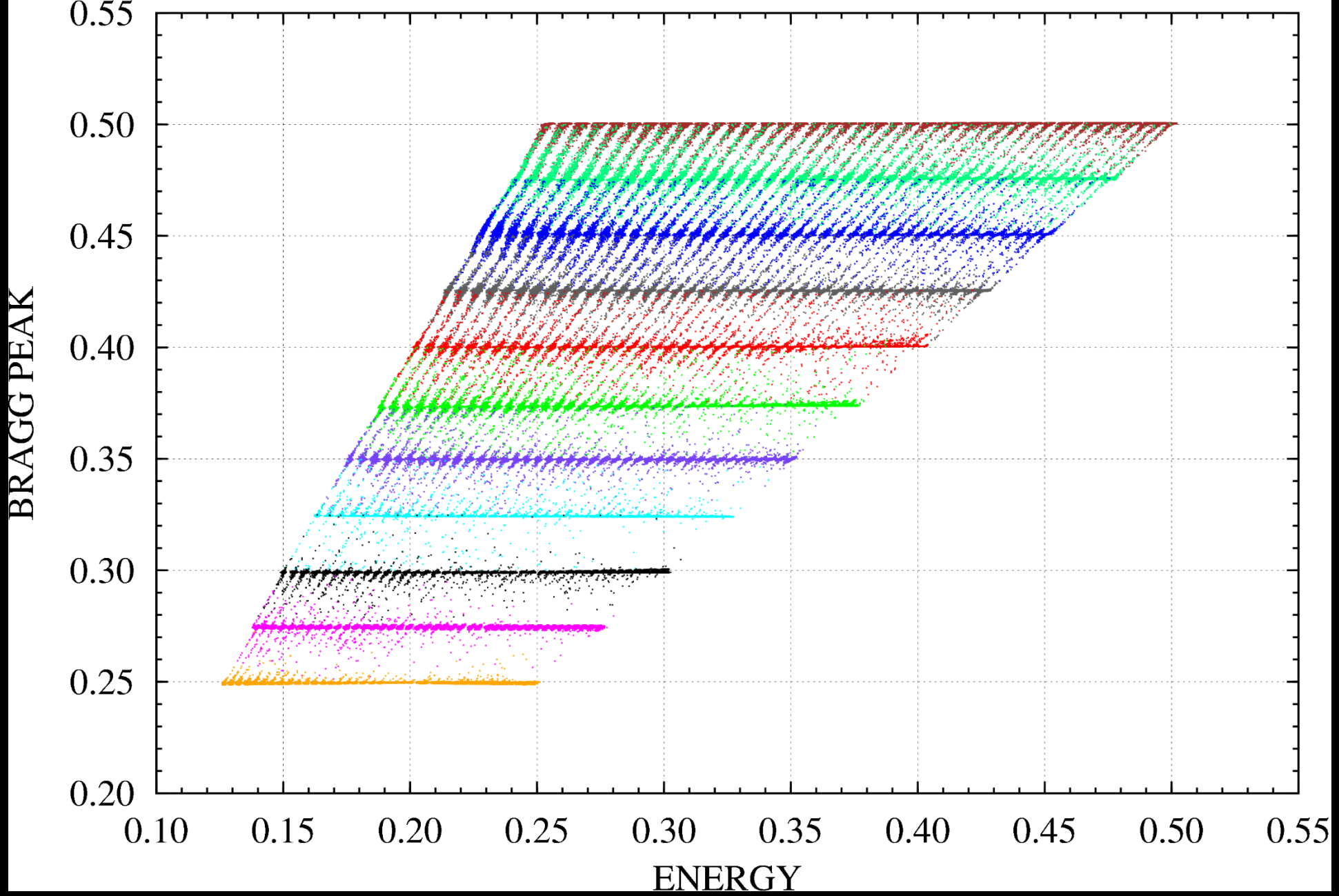


Fig. 2b. Scatter plot for a backpropagation learning algorithm using η = 0.3, α = 0.15 and a datasets with 451,000 BCs. This scatter plot is obtained after 176,000 training epochs.

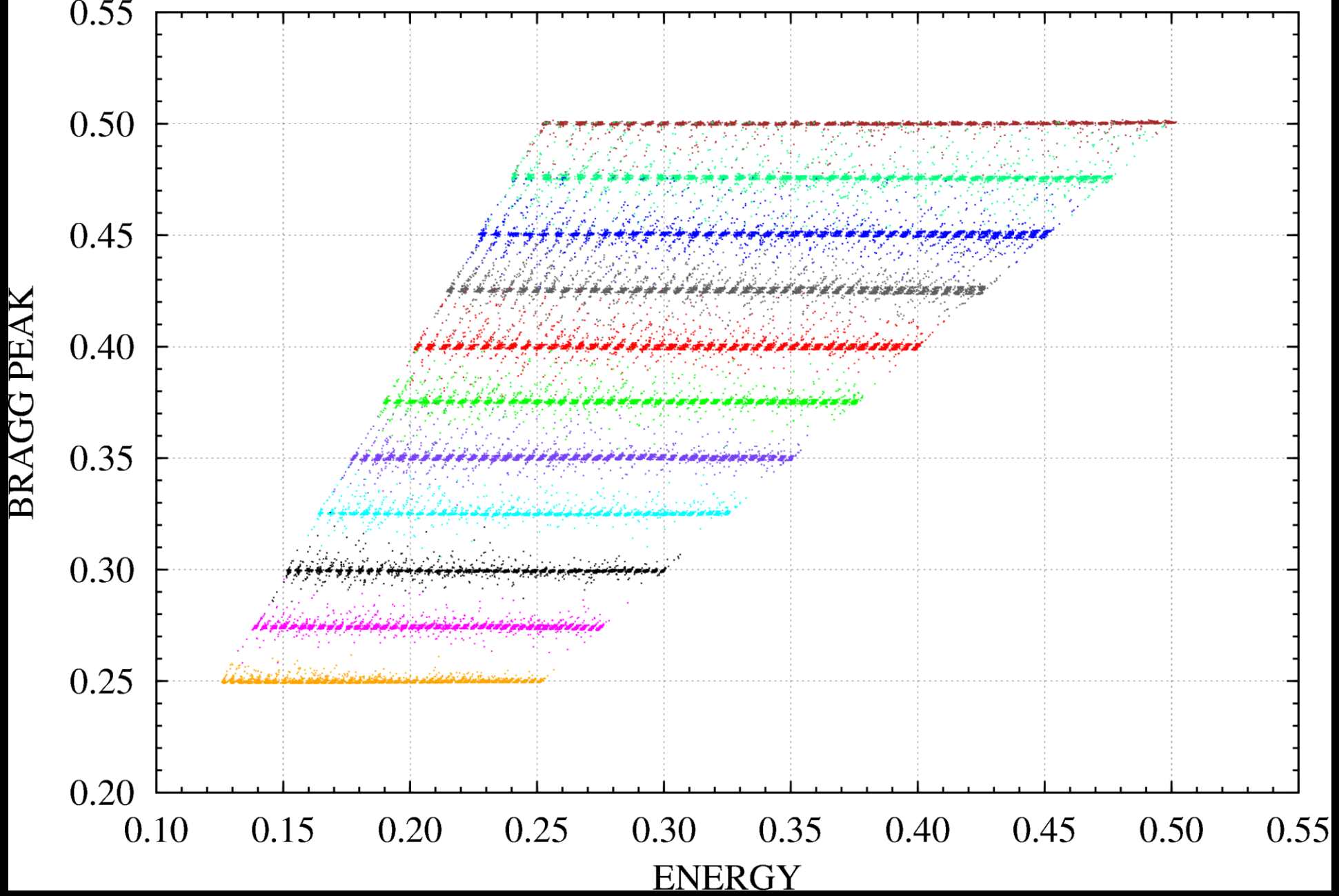


Fig. 2c. Scatter plot for a backpropagation learning algorithm using a weight decay term, η = 0.3, and a datasets with 45,100 BCs. This scatter plot is obtained after 3,330,000 training epochs.

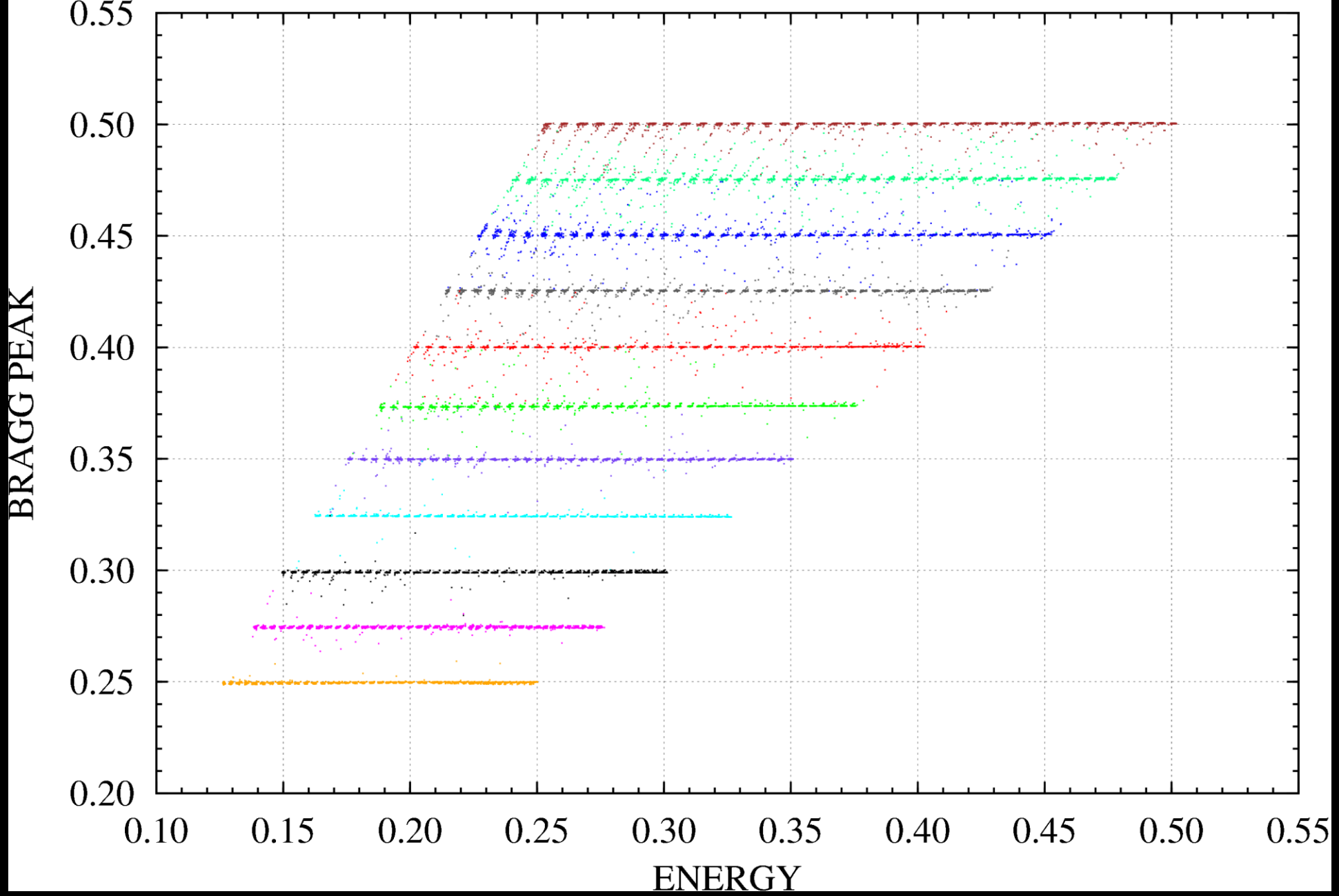


Fig. 2d. Scatter plot for a backpropagation learning algorithm using η = 0.3, α = 0.15 and a datasets with 451,000 BCs. This scatter plot is obtained after 176,000 training epochs, displaying only 100 point per class.

In Figs. 3a-d, we show the scatter plots corresponding to Figs. 2a-d, but now we display only the four classes corresponding to the two largest Bragg Peaks and the two largest energies. Again, in Fig. 3-d, we display only 100 point per class. In these plots one can easily observe the amount of class overlapping among the four classes displayed. One can easily verify that the smallest overlapping occurs in case 2, and also, the most compacted classes correspond to case 2.

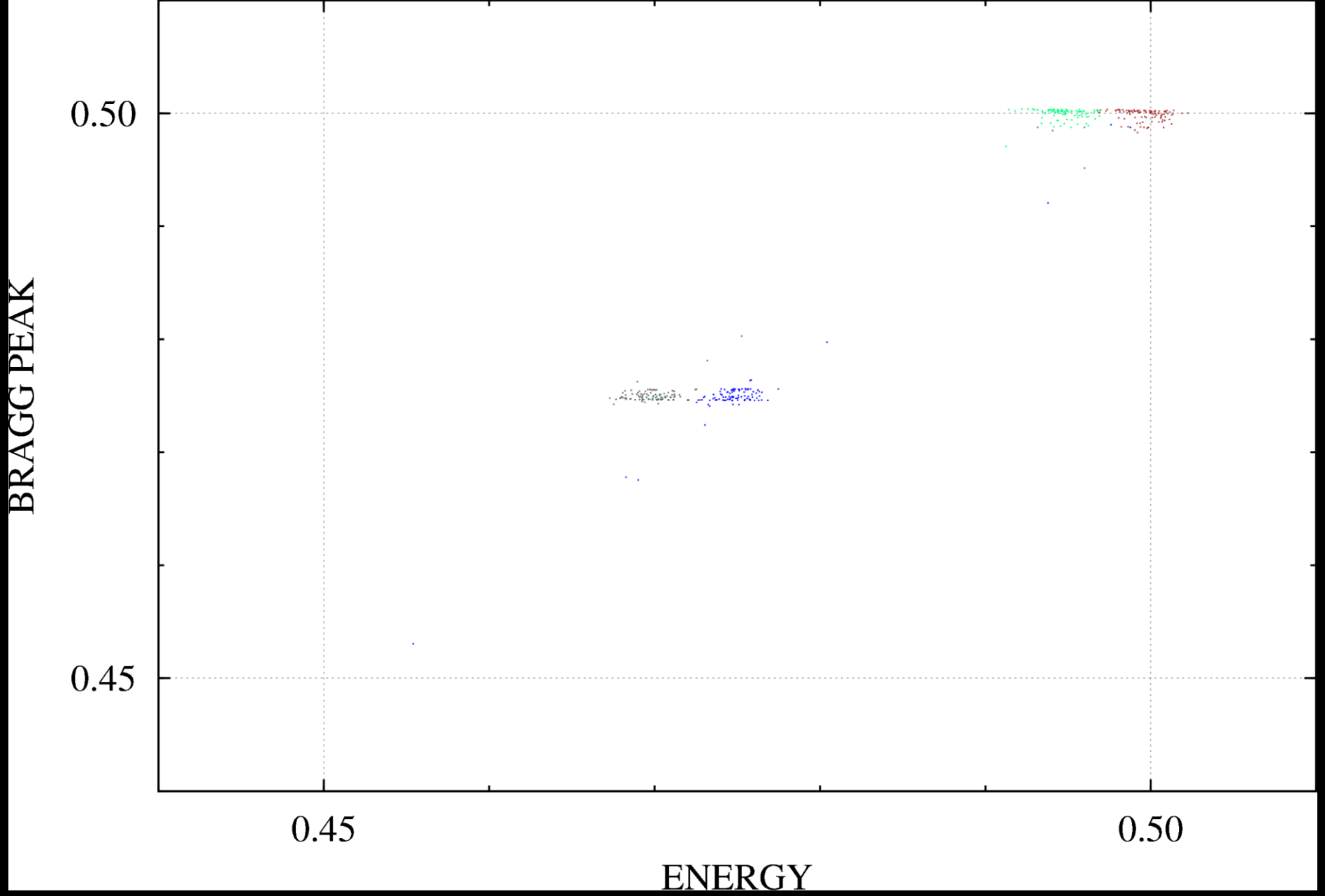


Fig. 3a. Scatter plot for a backpropagation learning algorithm using η = 0.3, α = 0.15 and a datasets with 45,100 BCs. This scatter plot is obtained after 196,000 training epochs.

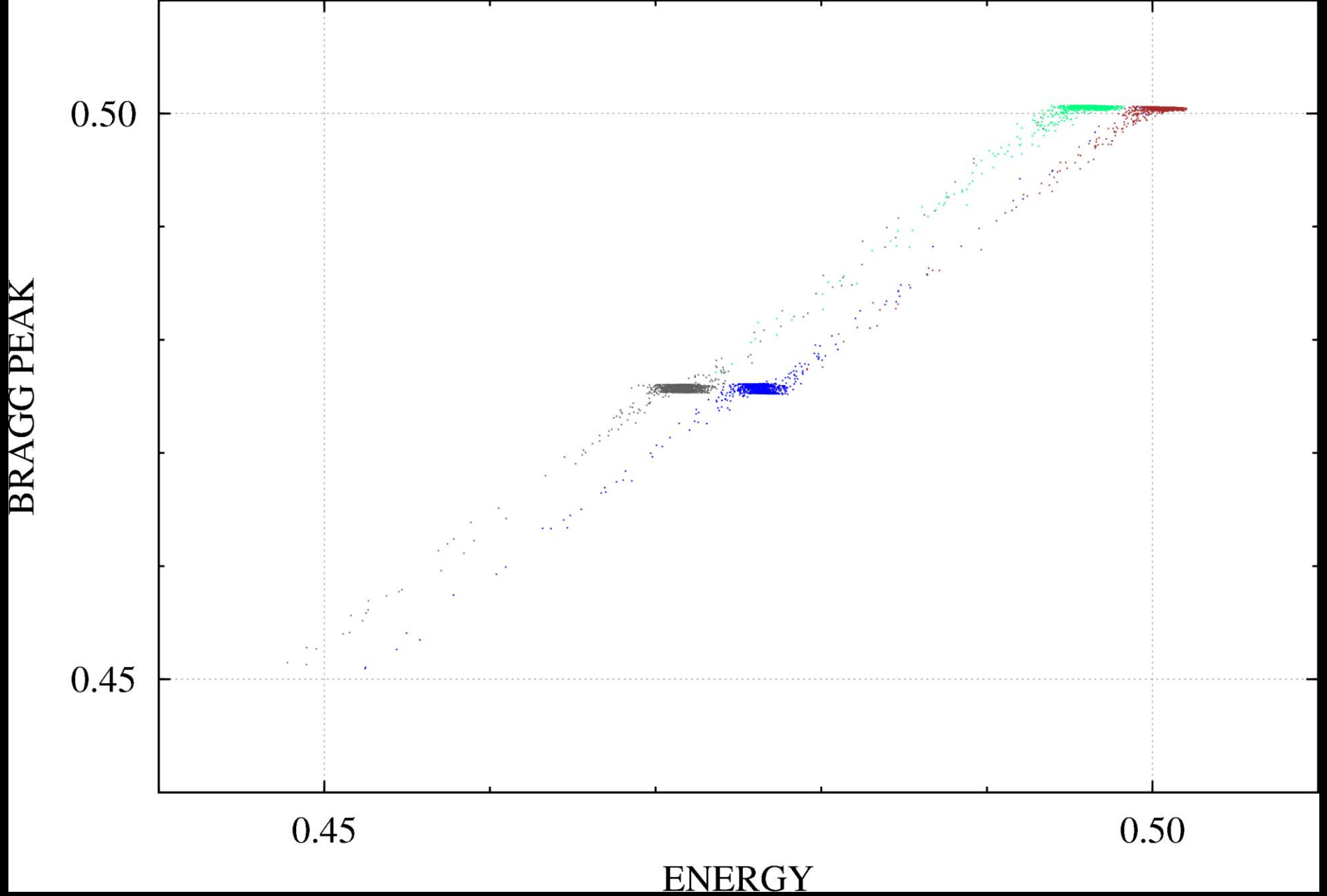


Fig. 3b. Scatter plot for a backpropagation learning algorithm using η = 0.3, α = 0.15 and a datasets with 451,000 BCs. This scatter plot is obtained after 176,000 training epochs.

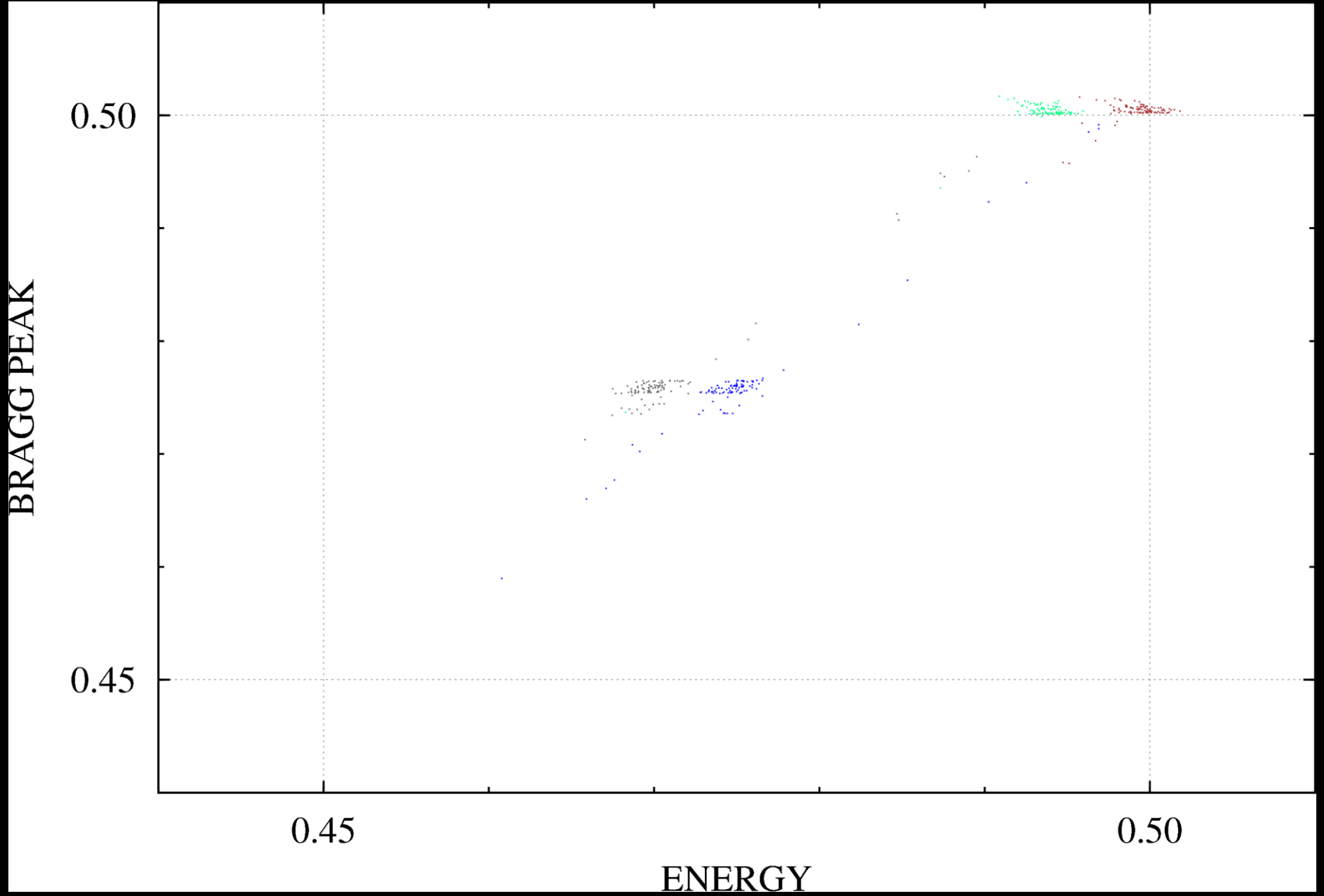


Fig. 3c. Scatter plot for a backpropagation learning algorithm using a weight decay term, η = 0.3, and a datasets with 45,100 BCs. This scatter plot is obtained after 3,330,000 training epochs.

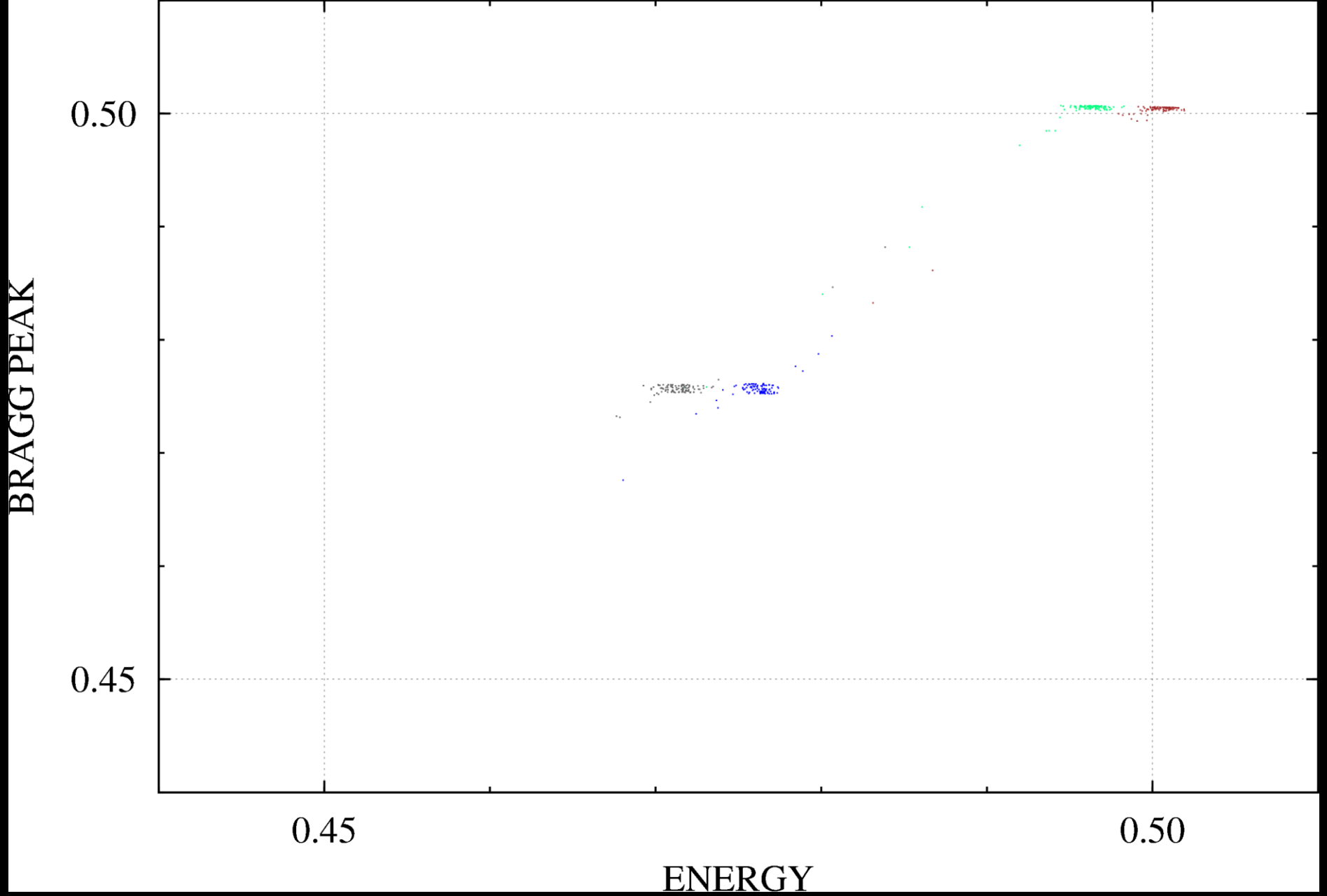


Fig. 3d. Scatter plot for a backpropagation learning algorithm using η = 0.3, α = 0.15 and a datasets with 451,000 BCs. This scatter plot is obtained after 176,000 training epochs , displaying only 100 point per class.

In Figs. 4a-c we present the curves of $|\vec{\omega}|$ vs the number of training epochs. In these figures we show the value of $|\vec{\omega}|$ at the number of training epochs where the minimum of the validation error curve is reached.

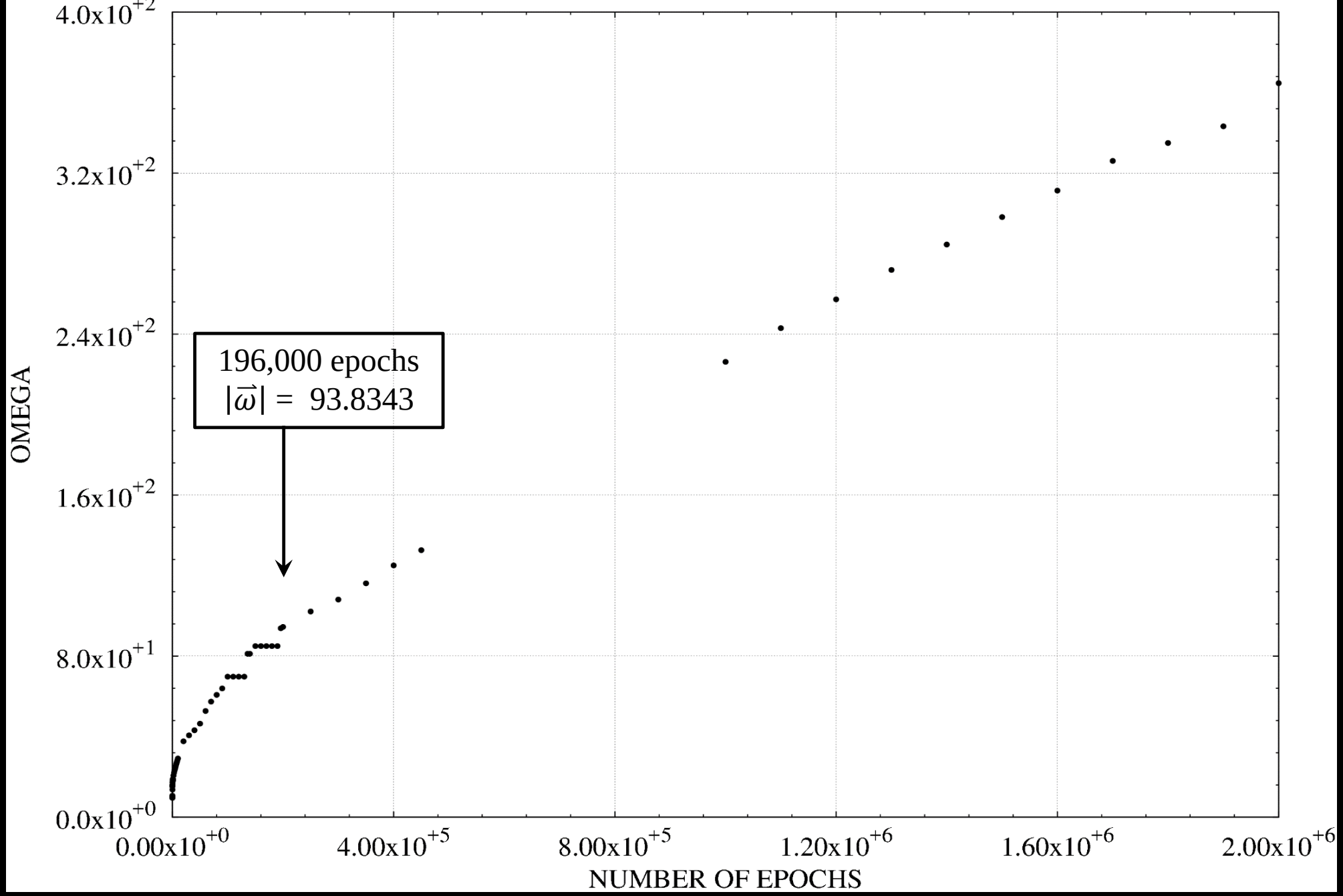


Fig. 4a. $|\vec{\omega}|$ vs number of epochs for $\eta = 0.3$, $\alpha = 0.15$ and 45,100 patterns in D^T and D^V.

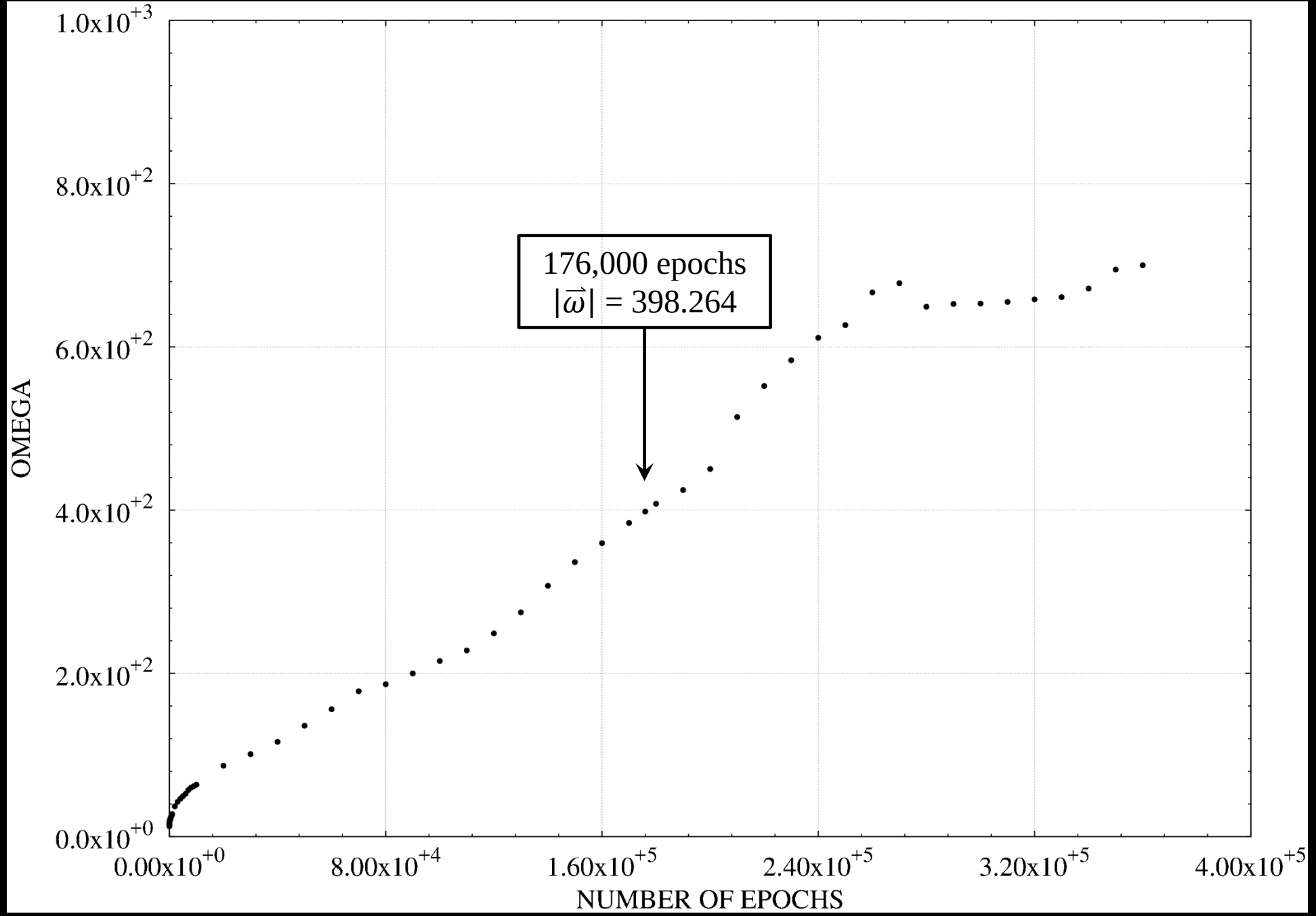


Fig. 4b. $|\vec{\omega}|$ vs number of epochs for $\eta = 0.3$, $\alpha = 0.15$ and 451,000 patterns in D^T and D^V.

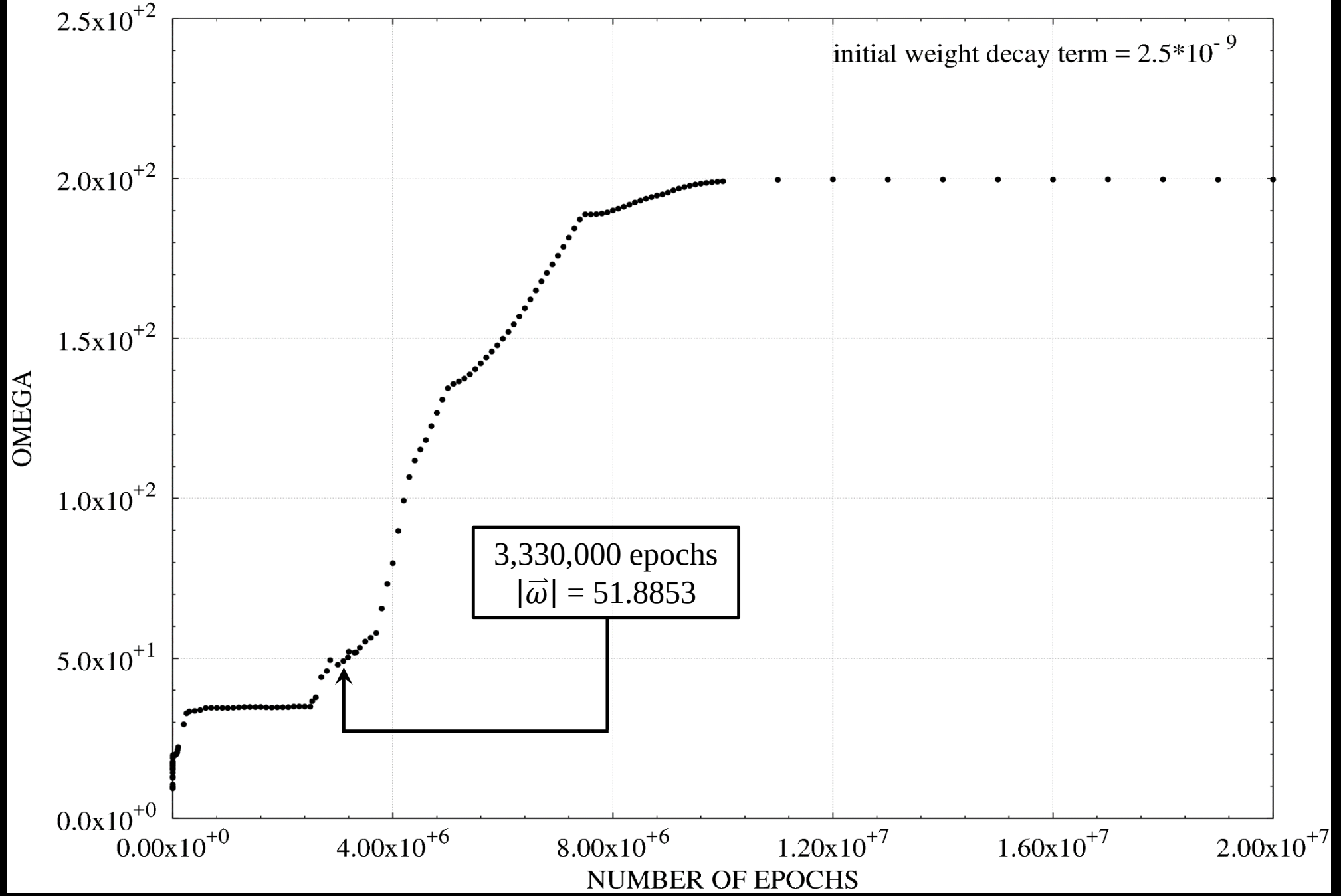
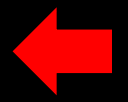


Fig. 4c. $|\vec{\omega}|$ vs number of epochs with weight decay, for $\eta = 0.3$ and 45,100 patterns in D^T and D^V.

Table 2. Absolute values of the weight array ($|\vec{w}(\rho)|$) at the minimum of the $E^V(\rho)$ curve for the three studied cases

case	number of epochs = ρ	$\|\vec{\omega}(\rho_{min})\|$
1	196,000	93.8343
2	176,000	398.264
3	3,330,000	51.8853



a large value !

A large $|\vec{w}(\rho)|$ implies a rather complex system. But the fact that the training dataset is a huge dataset (451,000 BCs), makes it quite difficult for the ANN to learn or to adapt its synaptic weight array value to the noise present in all the patterns. So the ANN has the chance to use is growing complexity to learn the small and subtle pattern features rather than the concomitant noise.

So, in the second case, the huge size of the dataset allows the ANN to reach a smaller minimum value for the validation error curve, learning subtle features present in the patterns instead of leaning the noise present in them.

1	$\langle E^V \rangle$(	186,000 →	206,000)	= $8.723767 \times 10^{-1} \pm 2.778259 \times 10^{-2}$	+ 28.3%
2	$\langle E^V \rangle$(	166,100 →	186,100)	= $6.254337 \times 10^{-1} \pm 2.702884 \times 10^{-2}$	
3	$\langle E^V \rangle$(	3,319,600 →	3,339,600)	= $7.877160 \times 10^{-1} \pm 2.483593 \times 10^{-2}$	+ 20.6%

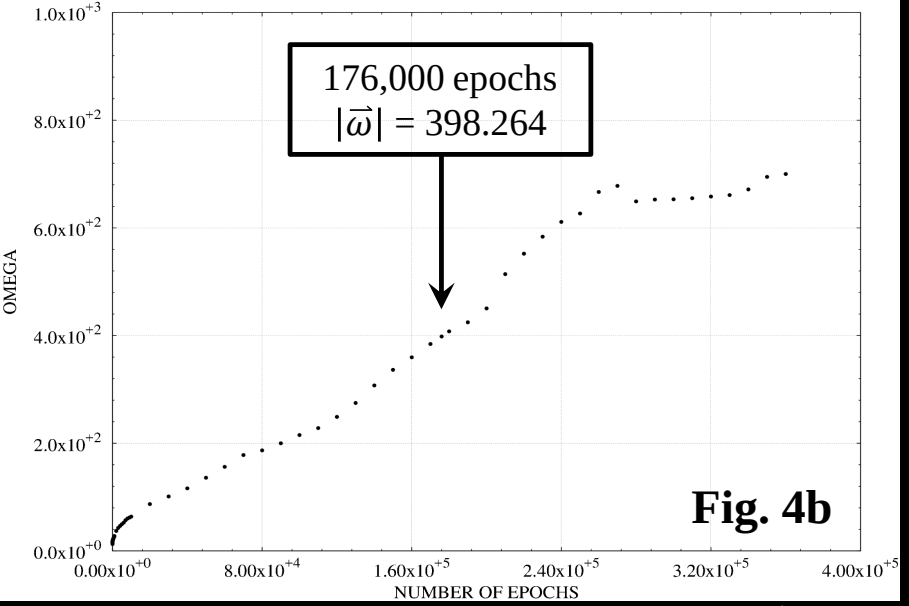


Fig. 4b

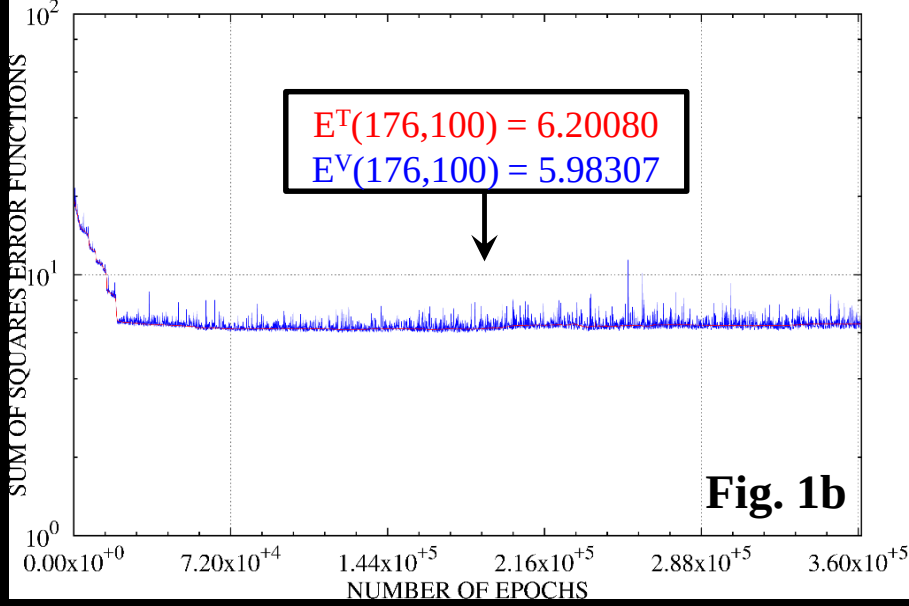


Fig. 1b

In Fig 4b, we see the $|\vec{\omega}(\rho)|$ value keeps on growing after 176,000 epochs, and the training and validation error curves remain close together, Fig. 1b, indicating the absence of data overfitting and overtraining.

Noise complexity grows

Ideal BCs complexity remains fixed

Dataset 45,100 BC

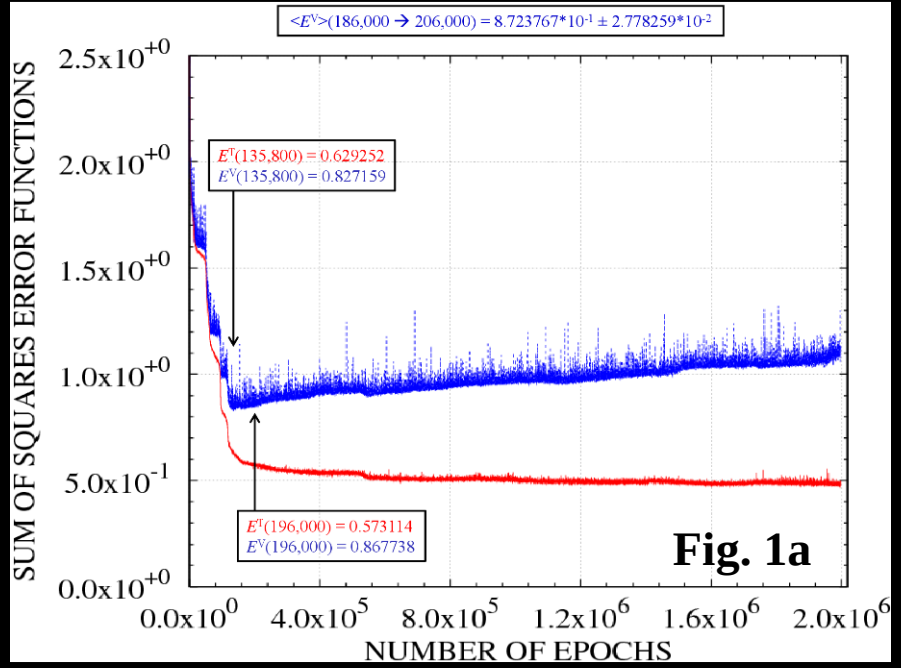
Ideal BCs complexity << noise complexity

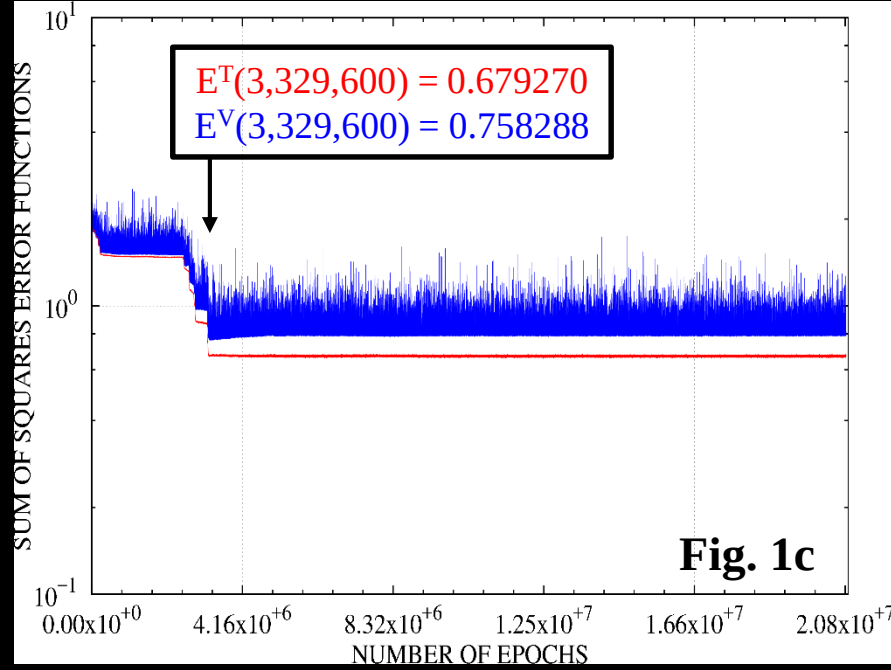
Dataset 451,000 BC

∴ With 451,000 BCs, the ANN complexity will be used to extract the true features present in the BCs rather than noise.

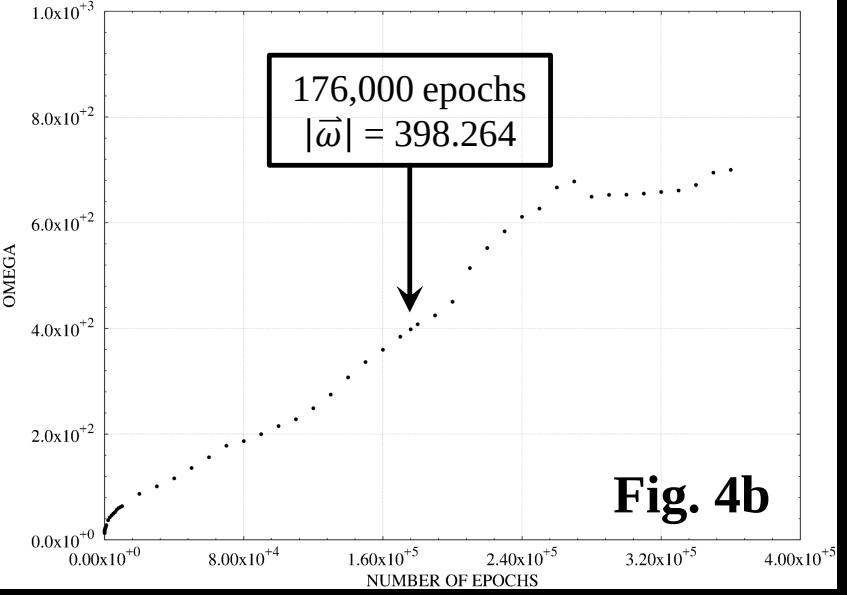
Table 2

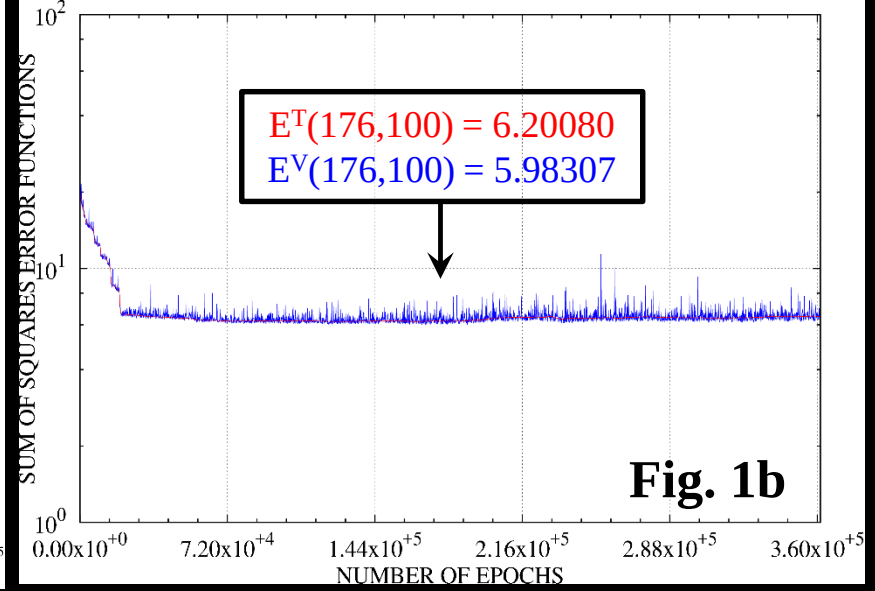
case	number of epochs = ρ	$\lvert\vec{\omega}(\rho_{min})\rvert$
1	196,000	93.8343
2	176,000	398.264
3	3,330,000	51.8853

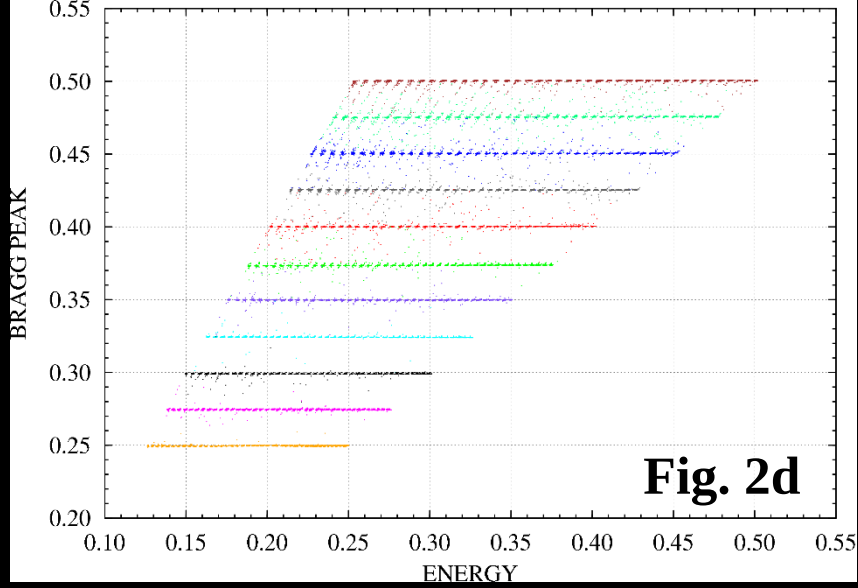




When comparing the $|\vec{w}(\rho_{min})|$ values of the first and the third cases, we see that it is larger for the first case. That means the weight decay term helps the model complexity of the ANN to adapt itself to the small and subtle features rather than to the noise, preventing or at least diminishing the occurrence of overfitting, i.e., learning noise. This overfitting preventing or diminishing occurrence is clearly observe when comparing the corresponding error curves.







The uncontrolled growth of the $E^V(\varrho)$ curve is also present in some extent in our best option, Fig. 4b. Nevertheless, its $E^T(\varrho)$ and $E^V(\varrho)$ error curves, Fig. 1b, remain close together up to 360,000 epochs, meaning that the ANN is not overtraining after it has reached its minimum value at 176,100 training epochs. explaining the better quality of its scatter plot, Fig. 2d, evaluated at its minimum.

DISCUSSION

In order to make a characterization of overfitting and overtraining it is convenient to define two sources of dataset complexity:

- **One source is associated with the ideal BC shape.**
- **The other is associated with the concomitant noise accompanying the BCs signals.**

If, at some point during the training process, it happens that the remaining unlearned features of both complexities are comparable, then, inevitably, overfitting will occur eventually. Once the minimum of the $E^V(\varrho)$ curve is reached, then overtraining will show up if training continues.

Since noise complexity grows as the dataset size grows, it happens that when the dataset is to big (451,000 BCs), i.e., noise complexity is larger than the one associated with the BC shape, then the ANN, during a considerable number of training epochs, will be able to learn many of the BC features without any chance of overfitting the noise component, because the ANN complexity is still not big enough to start learning the noise. In this case, overfitting will not show up.

Weight decay regularization

Learning the smaller subtle signal features will eventually require increasing the model complexity, i.e., the size of the synaptic weight array, which is precisely what the weight decay term controls. In the long run, the weight decay performance exceeds the simple back-propagation with early stopping alternative, but it is not capable of matching the results obtained when using a larger training dataset.

Summarizing

Learning small and subtle signal features requires two things:

a) **To increase the complexity of the system allowing the ANN to be able to learn those small and subtle signal features.**

b) **To provide the ANN the opportunity of using an increasing complexity to learn the small and subtle features rather than the noise that comes along with the signal.**

To accomplish restraint a) we used a weight decay term to control the synaptic array growth rate. To implement restraint b) we increased the size of the training dataset by a factor of ten making noise learning more difficult.

Using a larger training dataset has two important advantages:

i) **To allow an effective growing of the ANN complexity in order to learn the small and subtle signal features.**

ii) **To prevent noise learning while the ANN learns the small and subtle features.**

Finally, two related issues relevant to the present analysis are: overtraining and overfitting. According to [Te-95], the overfitting problem refers to exceeding some optimal ANN size, while overtraining refers to exceeding the number of training epochs required to train an ANN and start destroying the predictive ability of the network.

CONCLUSIONS

Following [Mo-92, Bi-95, Bi-06], we emphasized the relevance of optimizing the complexity of the model in order to achieve the best generalization. In this respect, one has to be careful in deciding the best option to optimize the ANN complexity used to solve a specific task. But a relevant aspect, one should keep in mind, is that another way to warrant a better generalization capability of an ANN is to use a large training dataset. In fact, the model complexity (ANN size) and the size of the training dataset ought to be selected in a mutually constrained way, trying to get a reasonable generalization capability.

THANKS

References

[Bi-95] Bishop, C. M., Neural Networks for Pattern Recognition. Oxford University Press (1995).

[Bi-06] Bishop C. M., “Pattern Recognition and Machine Learning”, (2006) Springer .

[Gr-82] Gruhn C.R., et. al., Nucl. Instr. and Meth. 196 (1982) 33-40.

[Ha-96] Hagan M. T., Demuth H. B., Beale M. H., De Jesús O., Neural Network Design, (1996).

[Mo-84] Moroni A., et al., Nucl. Instr. and Meth. 225 (1984) 57-64.

[Mo-92] Moody, J. E,. “The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems”, in J. E. Moody, S. J. Hanson and R. P. Lippmann, eds., Advances in Neural Information Processing Systems 4, Morgan Kaufmann Publishers, San Mateo, CA, (1992) 847–854.

[Sa-95] Sarle W. S., “Stopped training and other remedies for overfitting”, in Proc. Of the 27th Symposium on Interface, (1995).

[Te-95] Tetko I. V., J. Chem. Inf. Comput. Sci. 35,5 (1995) 826-833.

[Ve-06] Vega J. J, et al., Nucl. Instr. and Meth. B243 (2006) 232-240.

[Vi-87] Vineyard M. F., et al., Nucl. Instr. and Meth. A255 (1987) 507-511.