A New Fast Forecasting Technique using High Speed Neural Networks

HAZEM M. EL-BAKRY
Faculty of Computer Science & Information Systems,
Mansoura University, EGYPT
helbakry20@yahoo.com

NIKOS MASTORAKIS
Dept. of Computer Science
Military Institutions of University Education
(MIUE) - Hellenic Academy, Greece

Abstract: Forecasting is an important issue for many different applications. In this paper, a new efficient forecasting technique is presented. Such technique is designed by using fast neural networks (FNNs). The new idea relies on performing cross correlation in the frequency domain between the input data and the input weights of neural networks. It is proved mathematically and practically that the number of computation steps required for the proposed fast forecasting technique is less than that needed by conventional neural-based forecasting. Simulation results using MATLAB confirm the theoretical computations. The proposed fast forecasting technique increases the prediction speed and at the same time does not affect the predication accuracy. It is applied professionally for erythemal ultraviolet irradiance prediction.

Keywords: Fast Neural Network; Cross Correlation, Frequency Domain, Combined Neural Classifiers, Information Fusion, erythemal UV irradiance; total ozone.

I. Introduction

Improving the speed of forecasting is very important and necessary for real time applications. In this paper, we concentrate on increasing the speed of forecasting during the prediction phase. This improvement is achieved by applying cross correlation in the frequency domain. The cross correlation is preformed between the whole input data and the weights of neural networks. This new idea increases the speed of the prediction process compared to normal implementation of forecasting in time domain.

It was proved that performing cross correlation in the frequency domain is faster than time domain [56]. By the words "fast cross correlation", it is meant that cross correlation is performed in the frequency domain. A general fast pattern detection model using fast cross correlation was presented in [32-54]. Fast cross correlation was applied successfully for many different applications. Fast sub-image detection was achieved using fast cross correlation. A fast searching algorithm for face/object detection using neural networks and fast cross correlation was presented in [36,38,42,44]. Very fast iris detection using fast cross correlation was described in [43]. A faster algorithm for pattern detection using fast cross correlation and image decomposition was presented in [42,46,47,49]. The fastest pattern detection was achieved by using fast cross correlation, image decomposition and parallel processors. Furthermore, real time fast code detection for communication applications using fast cross correlation was introduced in [35,40,41]. In addition, a new time delay artificial neural network was invented using fast cross correlation as presented in [50,52]. As well as, an interesting mathematical application by using fast cross correlation was introduced [32,48]. Moreover, an Internet application for fast searching on web pages using fast cross correlation was presented in [54]. Finally, high speed data processing using fast cross correlation was introduced in [53].

A marked increase in the incidence of skin cancers has been observed in fair-skinned populations worldwide since the early 1970s [8]. This is strongly associated with personal habits in relation to sun exposure. Overexposure to ultraviolet radiation is the primary environmental risk factor in the development of ultraviolet -related adverse health effects, which include diseases of the eye (e.g., cataract, lens capsule deformation, ocular melanoma, etc.), skin wrinkles, delayed tanning, sunburn, carcinoma and also molecular changes within the cells [3,24]. As a result, it is of great interest to assess the changes in the level of ultraviolet radiation reaching the Earth's surface. Because of the observed ultraviolet series are not sufficiently long and do not allow to determine appropriate statistical characteristics with desirable accuracy, many efforts have been carried out in inferring this information from other available data sets [5]. One of them has been the use of empirical and statistical models [1,6,9,10,11,19,22,23,60]. One of the limitations imposed by these correlations is that they show a latitude dependency. Radiative transfer models have been also used to estimate ultraviolet levels at locations where measurements are not available [2,21,23,28,30]. Results from these studies suggest that model calculations in most cases can reproduce ultraviolet irradiance with uncertainties comparable to those of the measurements [20].

Contrary to the mathematical routines mentioned above, artificial intelligence may provide a suitable technique to improve the accuracy of this prediction. In this research, the validity of two different neural methodologies to solve a particular kind of inverse problem is discussed. Erythemal ultraviolet irradiance and a few number of sources are considered. The first methodology refers to the long-utilized multi-layer perceptron (MLP) neural networks whose performances are well known. The second methodology refers to the relatively new approach based on the bootstrap aggregating training algorithm. Results have been compared with the existing relevant data from three Egyptian sites in order to verify the validity of the methods. The atmosphere of Cairo (30° 05′ N, 32° 17′ E, 36 m) was chosen for the polluted case. On the other hand, Aswan (23°58′ N, 32° 47′ E, 192 m) and Mersa-Matruh (31°20′ N, 27°13′ E, 38 m) were chosen for the clarity of the atmosphere. Assessing of Erythemal Ultraviolet Irradiance is discussed in section II. Forecasting by using neural networks is described in section III. A new proposed forecasting algorithm by using fast neural networks is presented in section IV.

II. Assessing the Erythemal Ultraviolet Irradiance

Because of the apparent lack of long-term ultraviolet measurements, alternative methods for assessing the ultraviolet levels of the past have to be considered. Satellite retrieved ultraviolet is an important contributor in this area [13,31]. A global monitoring of surface ultraviolet can be done by using satellite measurements as from the total ozone mapping spectrometer (TOMS). Taking advantage of the TOMS satellite data availability, the required data for the models were obtained from NASA's website [58]. The TOMS' data have a daily global coverage over 1°x1.25° (latitude by longitude) grids. The total relative uncertainty in the radiance calibration is estimated to be 63% (though somewhat higher at high latitudes). For more detailed descriptions of the different sources of uncertainty the reader is referred to [8,59]. The collected data include the local noon erythemal Ultraviolet, reflectivity, ozone content, solar zenith angle, and aerosol and SO2 indices. This research was developed using 6-years data archive (1997–2002). This data was divided into two subsets: the training subset (comprising three complete years cover the period 1997-1999) and the performance subset (comprising the existing relevant data from 2000 to 2002). The models were built with the former data subset and they were, subsequently, verified with the latter data subset (i.e., the testing set was not included as a part of the training set). As a result, the high accuracy obtained demonstrated the ability of these techniques to produce accurate estimates. The monitoring sites have been chosen to cover the coastal and interior areas of Egypt.

III. Forecasting using Neural Networks

Artificial neural network (ANN) is a mathematical model, which can be set one or more layered and occurred from many artificial neural cells [16] Jang et al., 1997). The wide usage of the ANN may be due to the three basic properties: (1) the ability of the ANN as a parallel processing of the problems, for which if any of the neurons violate the constraints would not affect the overall output of the problem; (2) the ability of the ANN to extrapolate from historical data to generate forecasts; and (3) the successful application of the ANN to solve non-linear problems (Sazi, 2006). The history and theory of the ANN have been described in a large number of published literatures and will not be covered in this paper except for a very brief overview of how neural networks operate.

The ANN computation can be divided into two phases: learning phase and testing phase. The learning phase forms an iterative updating of the synoptic weights based upon the error back propagation algorithm. Back propagation algorithm is generalized of least mean square learning rule, which is an approximation of steepest descent technique. To find the best approximation, multi-layer feed forward neural network architecture with back propagation learning rule is used. A schematic diagram of typical multi-layer feed-forward neural network architecture is shown in Fig. 1. The network has five inputs and one output neuron in its linear output layer. The number of neurons in the hidden layer is varied to give the network enough power to solve the problem. Each neuron computes a weighted sum of the individual inputs ($I_1, I_2, \ldots, Ij$) it receives and adding it with a bias (b) to form the net input (x). The bias is included in the neurons to allow the activation function to be offset from zero.

$$sum = w_{1,1}.I_1 + w_{1,2}.I_2 + \ldots + w_{1,j}.I_j + b \qquad (1)$$

Where, wji is the connection weight between neuron j and neuron i. The net input (sum) is then passed to the subsequent layer through a non linear sigmoid function to form its own output (yj).

$$y_j = 1/\left(1 + e^{-sum}\right) \qquad (2)$$

Afterward, the output yj was compared with the target output tj using an error function of the form:

$$\delta_k = \left(t_j - y_j\right) y_j \left(1 - y_j\right) \qquad (3)$$

For the neuron in the hidden layer, the error term is given by the following equation:

$$\delta_j = y_j \left(1 - y_j\right) \sum_k \delta_k w_k \qquad (4)$$

where δk is the error term of the output layer, and wk is the weight between the hidden layer and output layer. The error was then propagated backward from the output layer to the input layer to update the weight of each connection at iteration (t + 1) as follows:

$$\begin{aligned} w_{ji}(t+1) &= w_{ji}(t) + \eta \delta_j \, y_j \\ &+ \alpha \left(w_{ii}(t) - w_{ii}(t-1)\right) \end{aligned} \qquad (5)$$

Choosing a small learning rate η leads to slow rate of convergence, and too large η leads to oscillation. The term α is called momentum factor and determines the effect of past weight changes on the current direction of movement. Both of these constant terms are specified at the start of the training cycle and determine the speed and stability of the network [18]. The process was repeated for each input pattern until the error was reduced to a threshold value.

The most challenging part of designing a predictive model is to come up with the best set of classifiers. The quest of selecting the best classifier is often difficult. Experimental results showed that the sets of misclassified samples from different classifiers don't usually overlap [12]. This observation motivated the idea of using multiple sets of classifiers to solve the problem of identification in the presence of noisy data [17]. A combination of different types of classifiers promises to lead to improved models compared to selecting one of the competitors [14]. Combining the decision of different classifiers means to fuse among the various outputs into a single decision as shown in Fig. 3. In this research, the bootstrap aggregating (*bagging*) classifier is used. The principle idea of *bagging* as originally described by [4] is to generate multiple independent classifiers by exploiting instability of classifier learning under changes to the learning set. In addition, one can exploit instability under different internal parameterization of the classifier [27]. In other world we can eliminate variance due to a particular choice of training set. The following pseudo-code describes the *bagging* technique introduced by Breiman [4].

```
Bagging (prediction algorithm A, dataset D, iterations T)
1) model generation
        for $i$ = 1 to T:
            generate a bootstrap sample D($i$) from D
            let M($i$) be result of training A on D($i$)
2) prediction for a given test instance $x$
        for i = 1 to T:
            let C(i) = output of M($i$) $= \sum_{i=1}^{T} \frac{M(i)}{T}$ on $x$        (6)
3) return class that appears most often among C(1)...C(T)
```

Given a standard training set D, we generate a bootstrap sample D (*i*). The T models are fitted using the above T bootstrap samples and combined by averaging the output. Thereafter, the predictions of *bagging* classifier are used as additional predictors for a multi-layer perceptron (MLP) network. An advantage of the MLP based *bagging* is its relative ease of interpretation, which can help the user to understand and improve the classification rules. After trying a number of different configurations, it was found that 10 neurons in the hidden layer with $\eta = 0.02$ and $\alpha = 0.8$ yield the best results. The performance is then evaluated by the overall prediction accuracy.

In order to check the performance of the combination of bagging algorithm with MLP network several error measures, averaged over the period of record, were computed for each site. Mean absolute errors (MAE), mean absolute percentage errors (MAPE), mean bias errors (MBE), standard errors (SE) and root mean square errors (RMSE) were computed. Additional estimator namely correlation coefficient, r, was used to test the linear relation between estimated and observed values (see appendix "A"). The frequency distributions of the difference between estimated and observed values were also introduced.

As can be seen in Figures 4, 5 and 6, the combination of bagging algorithm with MLP network has minimum error rates. It can be seen that, the MLP overall prediction accuracy is only 80.9%. When bagging algorithm is used, the accuracy reaches 94.8%; an improvement of about 13.9% was achieved. To be more precise, the prediction accuracies derived by MLP networks adjusted for Cairo, Aswan and Mersa-Matruh were found to be 70.9%, 88.4% and 83.3%, respectively which corresponds to 89.2%, 96.4% and 98.9% in the case of MLP based bagging algorithm. The relatively large error rates observed at Cairo may be attributed to air pollution which can prejudice its predictive accuracy. The ambient air in this area contains more UV absorbing materials such as aerosols, CO, SO2, NO2 and O3 than would be found at a site with pristine air conditions [7].

An analysis of the entries in Figures 4, 5 and 6 shows that the MBE is positive in some cases and negative in others suggesting that none of the two models over or under-estimates erythemal ultraviolet irradiance consistently. The tendency for overestimation was found only in the MLP application at Cairo site. As can be seen, the combination of MLP with bagging algorithm exhibited low MBE rates, implying that it has a good long-term representation of the physical problem. For instance, the range of MBE derived by the MLP network at Cairo site, was 13.5% to -0.8%, while the range of MBE for the MLP based bagging algorithm was 2.14% to -2.9%. The other two sites follow with relatively lower values.

Plots of the difference between predicted and observed values, given in Fig. 7, illustrate these biases. Inspecting the entries in Fig. 7, it may be observed that the distribution of MLP based bagging is quite different from the distribution of the MLP. In general, the distribution of MLP based bagging is symmetric at 0 while, the distribution of MLP is slightly skewed to the left. It has been found that 79.1% of the MLP based bagging dataset was estimated within a deviation of less than ±10 mW/m2 and 13.9% of these deviations lay within the range of 15 to 50 mW/m2. In the case of MLP, these values are about 65.6% and 26.8%, respectively. This finding indicate that the results obtained by MLP based bagging are generally satisfactory compared to the results of MLP.

It is also evident that, the RMSE obtained at Cairo was large, but remain in a domain of errors for which this approach can be applied with good accuracy. Over the entire period, the RMSE of the MLP network was found to be 29.1%; corresponding to 10.9% in the case of MLP based bagging algorithm. It can be observed that for all cities, the error rates are low in summer and then increased in the winter. It is not wholly obvious as to the reason for the higher error rates in winter, but it may have been the result of restricted air flows and low mixing depth, thus resulting in ineffective air pollution dispersion and dilution. At Aswan and Mersa-Matruh, the RMSE rates are 3.6% and 1.1%, respectively, which shows the good prediction ability of the MLP network even when bagging algorithm is presented.

To further test the performance of the two models SE, MAE, and MAPE were computed. Although the results are mixed meaning that, for most cases, the MLP yields the minimum SE rates, while on others the reverse is true, the overall SE of the MLP is about 10%, while that of the MLP based bagging is about 8.7%. The obtained results also show that the overall MAE rates reduce from 9% to 8% when bagging was used. In the application of MAPE, the overall rate derived by MLP was 6.5% compared with 4.7% for the combination of MLP with bagging algorithm. As a final point, the generalization of the models was tested by the correlation coefficient, r. In general, the two models provide a tighter fit for all sites, although the r appears to be lowest in Cairo and Aswan. Figures 6 and 7 shows this detail well. It can be easily seen from Figures 4 and 5 that, the r for Cairo and Aswan does not perform so well as that for Mersa-Matruh site; its mean value is 0.8 compared with 0.9 for Mersa-Matruh. These results show that at least 80% of the variation in the erythemal ultraviolet irradiance can be explained by the input parameters. Further investigation of this indicator indicated that, for both models, r is barely significant ($r \cong 0.66$) in summer. Strong emissions of trace gases by vehicles traveling on the cities' narrow roads and suspended soil dust are responsible for the low value observed for r at all sites. In [26] the authors reported that the reduction in the received UV radiation due to dust ranged between 33% and 59%. It's worthy noted here that, trace gases and suspended particles are not included in the training set. The main criterion was the availability of these factors regarding the period covered. From the relatively reasonable magnitude of errors, it can therefore be concluded that the combination of MLP with bagging algorithm can be recommended to predict the noontime erythemal ultraviolet irradiance in daily scale for all sites.

IV. A New Proposed Forecasting Algorithm by using FNNs

FNNs are shown in Fig. 2. Computing the resulted output; for a certain pattern of information; in the incoming serial data, is a prediction problem. First neural networks are trained to predict the erythemal ultraviolet irradiance and this is done in time domain. In pattern detection phase, each position in the incoming matrix is processed to predict the erythemal ultraviolet irradiance by using neural networks. At each position in the input one dimensional matrix, each sub-matrix is multiplied by a window of weights, which has the same size as the sub-matrix. The outputs of neurons in the hidden layer are multiplied by the weights of the output layer. Thus, we may conclude that the whole problem is a cross correlation between the incoming serial data and the weights of neurons in the hidden layer.

The convolution theorem in mathematical analysis says that a convolution of f with h is identical to the result of the following steps: let F and H be the results of the Fourier Transformation of f and h in the frequency domain. Multiply F and H* in the frequency domain point by point and then transform this product into the spatial domain via the inverse Fourier Transform. As a result, these cross correlations can be represented by a product in the frequency domain. Thus, by using cross correlation in the frequency domain, speed up in an order of magnitude can be achieved during the prediction process [32-54]. Assume that the size of the input data is 1xn. In the prediction phase, a sub matrix I of size 1xn (sliding window) is extracted from the tested matrix, which has a size of 1xN. Such sub matrix, which contains the input pattern, is fed to the neural network. Let W_i be the matrix of weights between the input sub-matrix and the hidden layer. This vector has a size of 1xn and can be represented as 1xn matrix. The output of hidden neurons h(i) can be calculated as follows:

$$h_i = g\left(\sum_{k=1}^{n} W_i(k)I(k) + b_i\right) \tag{7}$$

where g is the activation function and b(i) is the bias of each hidden neuron (i). Equation 7 represents the output of each hidden neuron for a particular sub-matrix I. It can be obtained to the whole input matrix Z as follows:

$$h_i(u)=g\left(\sum_{k=-n/2}^{n/2} W_i(k)\ Z(u+k)+b_i\right) \tag{8}$$

Equation 8 represents a cross correlation operation. Given any two functions f and d, their cross correlation can be obtained by [56]:

$$d(x)\otimes f(x)=\left(\sum_{n=-\infty}^{\infty} f(x+n)d(n)\right) \tag{9}$$

Therefore, Eq. 8 may be written as follows [32-54]:

$$h_i = g\left(W_i \otimes Z + b_i\right) \tag{10}$$

where h_i is the output of the hidden neuron (i) and h_i (u) is the activity of the hidden unit (i) when the sliding window is located at position (u) and (u) $\in$ [N-n+1].

Now, the above cross correlation can be expressed in terms of one dimensional Fast Fourier Transform as follows [32-54]:

$$W_i \otimes Z = F^{-1}\left(F(Z)\bullet F^*\left(W_i\right)\right) \tag{11}$$

Hence, by evaluating this cross correlation, a speed up ratio can be obtained comparable to conventional neural networks. Also, the final output of the neural network can be evaluated as follows:

$$O(u) = g\left(\sum_{i=1}^{q} W_O(i)\, h_i(u) + b_O\right) \qquad (12)$$

where q is the number of neurons in the hidden layer. O(u) is the output of the neural network when the sliding window located at the position (u) in the input matrix Z. W_o is the weight matrix between hidden and output layer.

The complexity of cross correlation in the frequency domain can be analyzed as follows:

1- For a processed matrix of 1xN elements, the 1D-FFT requires a number equal to $N\log_2 N$ of complex computation steps [55]. Also, the same number of complex computation steps is required for computing the 1D-FFT of the weight matrix at each neuron in the hidden layer.

2- At each neuron in the hidden layer, the inverse 1D-FFT is computed. Therefore, q backward and (1+q) forward transforms have to be computed. Therefore, for a given matrix under test, the total number of operations required to compute the 1D-FFT is $(2q+1)N\log_2 N$.

3- The number of computation steps required by FNNs is complex and must be converted into a real version. It is known that, the one dimensional Fast Fourier Transform requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions [55]. Every complex multiplication is realized by six real floating point operations and every complex addition is implemented by two real floating point operations. Therefore, the total number of computation steps required to obtain the 1D-FFT of a 1xN matrix is:

$$\rho=6((N/2)\log_2 N) + 2(N\log_2 N) \qquad (13)$$

which may be simplified to:

$$\rho=5N\log_2 N \qquad (14)$$

4- Both the input and the weight matrices should be dot multiplied in the frequency domain. Thus, a number of complex computation steps equal to qN should be considered. This means 6qN real operations will be added to the number of computation steps required by FNNs.

5- In order to perform cross correlation in the frequency domain, the weight matrix must be extended to have the same size as the input matrix. So, a number of zeros = (N-n) must be added to the weight matrix. This requires a total real number of computation steps = q(N-n) for all neurons. Moreover, after computing the FFT for the weight matrix, the conjugate of this matrix must be obtained. As a result, a real number of computation steps = qN should be added in order to obtain the conjugate of the weight matrix for all neurons. Also, a number of real computation steps equal to N is required to create butterflies complex numbers ($e^{-jk(2\Pi n/N)}$), where 0<K<L. These (N/2) complex numbers are multiplied by the elements of the input matrix or by previous complex numbers during the computation of FFT. To create a complex number requires two real floating point operations. Thus, the total number of computation steps required for FNNs becomes:

$$\sigma=(2q+1)(5N\log_2 N) +6qN+q(N-n)+qN+N \qquad (15)$$

which can be reformulated as:

$$\sigma=(2q+1)(5N\log_2 N)+q(8N-n)+N \qquad (16)$$

6- Using sliding window of size 1xn for the same matrix of 1xN pixels, q(2n-1)(N-n+1) computation steps are required when using conventional neural networks (CNNs) for certain pattern detection or processing (n) input data. The theoretical speed up factor η can be evaluated as follows:

$$\eta = \frac{q(2n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \qquad (17)$$

Neural networks accept serial input data with fixed size (n). Therefore, the number of input neurons equals to (n). Instead of treating (n) inputs, our new approach is to collect all the input data together in a long vector (for example 100xn). Then the input data is tested by neural networks as a single pattern with length L (L=100xn). Such a test is performed in the frequency domain. Complex-valued neural networks mean that the inputs, weights, thresholds and the activation function have complex values [57]. In this section, formulas for the speed up ratio with different types of inputs will be presented. The special case of only real input values (i.e. imaginary part=0) will be considered. Also, the speed up ratio in the case of a one and two dimensional input matrix will be concluded. The operation of FNNs depends on computing the Fast Fourier Transform for both the input and weight matrices and obtaining the resulting two matrices. After performing dot multiplication for the resulting two matrices in the frequency domain, the Inverse Fast Fourier Transform is calculated for the final matrix. Here, there is an excellent advantage with FNNs that should be mentioned. The Fast Fourier Transform is already dealing with complex numbers, so there is no change in the number of computation steps required for FNNs. Therefore, the speed up ratio in the case of complex-valued neural networks can be evaluated as follows:

1) In case of real inputs

A) For a one dimensional input matrix

Multiplication of (n) complex-valued weights by (n) real inputs requires (2n) real operations. This produces (n) real numbers and (n) imaginary numbers. The addition of these numbers requires (2n-2) real operations. Therefore, the number of computation steps required by conventional neural networks can be calculated as:

$$\theta=q(2n-1)(N-n+1) \qquad (18)$$

The speed up ratio in this case can be computed as follows:

$$\eta = \frac{2q(2n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \quad (19)$$

The theoretical speed up ratio for searching short successive (n) data in a long input vector (L) using complex-valued neural networks is shown in Figures 8, 9, and 10. Also, the practical speed up ratio for manipulating matrices of different sizes (L) and different sized weight matrices (n) using a 700 MHz processor and MATLAB is shown in Fig. 11.

B) For a two dimensional input matrix

Multiplication of (n^2) complex-valued weights by (n^2) real inputs requires ($2n^2$) real operations. This produces (n^2) real numbers and (n^2) imaginary numbers. The addition of these numbers requires ($2n^2$-2) real operations. Therefore, the number of computation steps required by conventional neural networks can be calculated as:

$$\theta=2q(2n^2-1)(N-n+1)^2 \quad (20)$$

The speed up ratio in this case can be computed as follows:

$$\eta = \frac{2q(2n^2-1)(N-n+1)^2}{(2q+1)(5N^2\log_2 N^2)+q(8N^2-n^2)+N} \quad (21)$$

The theoretical speed up ratio for detecting (nxn) real valued submatrix in a large real valued matrix (NxN) using complex-valued neural networks is shown in Figures 12, 13, and 14. Also, the practical speed up ratio for manipulating matrices of different sizes (NxN) and different sized weight matrices (n) using a 700 MHz processor and MATLAB is shown in Fig. 15.

2) In case of complex inputs

A) For a one dimensional input matrix

Multiplication of (n) complex-valued weights by (n) complex inputs requires (6n) real operations. This produces (n) real numbers and (n) imaginary numbers. The addition of these numbers requires (2n-2) real operations. Therefore, the number of computation steps required by conventional neural networks can be calculated as:

$$\theta=2q(4n-1)(N-n+1) \quad (22)$$

The speed up ratio in this case can be computed as follows:

$$\eta = \frac{2q(4n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \quad (23)$$

The theoretical speed up ratio for searching short complex successive (n) data in a long complex-valued input vector (L) using complex-valued neural networks is shown in Figures 16, 17, and 18. Also, the practical speed up ratio for manipulating matrices of different sizes (L) and different sized weight matrices (n) using a 700 MHz processor and MATLAB is shown in Figure 19.

B) For a two dimensional input matrix

Multiplication of (n^2) complex-valued weights by (n^2) real inputs requires ($6n^2$) real operations. This produces (n^2) real numbers and (n^2) imaginary numbers. The addition of these numbers requires ($2n^2$-2) real operations. Therefore, the number of computation steps required by conventional neural networks can be calculated as:

$$\theta=2q(4n^2-1)(N-n+1)^2 \quad (24)$$

The speed up ratio in this case can be computed as follows:

$$\eta = \frac{2q(4n^2-1)(N-n+1)^2}{(2q+1)(5N^2\log_2 N^2)+q(8N^2-n^2)+N} \quad (25)$$

The theoretical speed up ratio for detecting (nxn) complex-valued submatrix in a large complex-valued matrix (NxN) using complex-valued neural networks is shown in Figures 20, 21, and 22. Also, the practical speed up ratio for manipulating matrices of different sizes (NxN) and different sized weight matrices (n) using a 700 MHz processor and MATLAB is shown in Fig. 23.

In practical implementation, the multiplication process consumes more time than the addition one. The effect of the number of multiplications required for conventional neural networks in the speed up ratio is more than the number of multiplication steps required by the faster neural networks. Also, the variations in Pc clock have an effect on practical computations.

For a one dimensional matrix, from Figures 8,9,10,11,16,17,18 and 19, we can conclude that the response time for vectors with short lengths are faster than those which have longer lengths. For example, the speed up ratio for the vector of length 10000 is faster that of length 1000000. The number of computation steps required for a vector of length 10000 is much less than that required for a vector of length 40000. So, if the vector of length 40000 is divided into 4 shorter vectors of length 10000, the number of computation steps will be less than that required for the vector of length 40000. Therefore, for each application, it is useful at the first to calculate the optimum length of the input vector. The same conclusion can be drawn in case of processing the two dimensional input matrix as shown in Figures 12,13,14,15,20,21,22, and 23. From these Figures, it is clear that the maximum speed up ratio is achieved at image size (N=200) when n=20, then at image size (N=300) when n=25, and at image size (N=400) when n=30. This confirms the previous results presented in [42] on fast subimage detection based on neural networks and image decomposition. Using this technique, it was proved that the speed up ratio of neural networks becomes faster when the input image is divided into many subimages and each subimage is processed in the frequency domain separately using a single fast neural processor. Another point of interest should be noted. In CNNs, if the whole input data (N) is available, then there is a waiting time for each group of (n) input data so that conventional neural networks can release their output for the previous group of (n) data. In contrast, FNNS can process the total N data directly with zero waiting time. For example, if the total (N) input data is appeared at the input neurons, then:

1- CNNs can process only data of size (n) as the number of input neurons = (n).
2- The first group of (n) data is processed by CNNs.
3- The second group of (n) data must wait for a waiting time = τ, where τ is the response time consumed by CNNs for treating each group of (n) input data.
4- The third group of (n) data must wait for a waiting time = 2τ corresponding to the total waiting time required by CNNs for treating the previous two groups.
5- The fourth (n) data must wait for a waiting time = 3τ.
6- The last group of (n) data must wait for a waiting time = (N-n)τ.

As a result, the wasted waiting time in the case of CNNs is (N-n)τ. In the case of FNNS, there is no waiting time as the whole input data (Z) of length (N) will be processed directly and the time consumed is the only time required by FNNs themselves to produce their output.

V. Conclusion

A new fast neural-based forecasting technique has been presented. Theoretical computations have shown that the proposed fast forecasting technique require fewer computation steps than conventional one. This has been achieved by applying cross correlation in the frequency domain between the input data and the input weights of neural networks. Simulation results have confirmed this proof by using MATLAB. This algorithm can be successfully applied to any application that uses time delay neural networks. In addition, the bagging algorithm has been used to improve the prediction accuracy reported by the MLP network. It has been found that, the daily errors associated with MLP based bagging algorithm are consistently smaller and less variable than those obtained by using the MLP network. These results have demonstrated the efficiency of the bagging algorithm together with FNNs to estimate the erythemal ultraviolet irradiance.

Appendix "A"

One of the most common indicators used in error analysis is the mean absolute error. This term is used similar to variance. The MAE of an estimator yj with respect to the estimated parameter xj is defined as:

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - x_j\right| \tag{26}$$

where, n is the number of data points. The MAPE is measure of accuracy in a fitted time series.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{j=1}^{n}\left|\frac{y_j - x_j}{y_j}\right| \tag{27}$$

We used the MBE to describe how much the predictor underestimates or overestimates the situation. The MBE was determined using the following equation:

$$\mathrm{MBE} = \frac{\sum_{j=1}^{n}\left(\frac{y_j - x_j}{x_j}\right)}{n} \tag{28}$$

The SE is the standard deviation of the sampling distribution and may be estimated by the formula $\frac{\sigma}{\sqrt{n}}$ where σ is the standard deviation of the population distribution. The mean squared error (MSE) of an estimator is the square of the amount by which the estimator differs from the quantity to be estimated. The difference occurs because the estimator doesn't account for information that could produce a more accurate estimate. The RMSE which gives an idea of the magnitude of the non-systematic error is then simply defined as the square root of the MSE. The mathematical formula of the RMSE is given by:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{J=1}^{n}\left(y_j - x_j\right)^2}{n}} \tag{29}$$

In general, correlation coefficient, r, indicates the strength and direction of a linear relationship between two random variables. The correlation is 1 in the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, and some value in between in all other cases, indicating the degree of linear dependence between the variables.

$$r = \frac{\sum_{j}\left(y_j - \bar{y}\right)\cdot\left(x_j - \bar{x}\right)}{\left\{\left[\sum_{j}\left(y_j - \bar{y}\right)^2\right]\cdot\left[\sum_{j}\left(x_j - \bar{x}\right)^2\right]\right\}^{0.5}} \tag{30}$$

Where, $\bar{x}$ is the observed mean value and $\bar{y}$ is the predicted mean value.

References

[[1] S. Al-Aruri, "The empirical relationship between global radiation and global ultraviolet (0.290–0.385 lm) solar radiation components," Solar Energy vol. (45), no. (2), 1990, pp. 61–64.

[2] D. S. Balis, C. S. Zerefos, K. Kourtidis, A. F. Bais, A. Hofzumahaus, A. Kraus, R. Schmitt, M. Blumthaler, and G. P. Gobbi, "Measurements and modeling of photolysis rates during the photochemical activity and ultraviolet

radiation (PAUR) II campaign," Journal of Geophysical Research vol. (107), no. (D18), 2002, p 8138.

[3] D.S. Berger, F. Urbach, "A climatology of sun burning ultraviolet," Journal of Photochemistry and Photobiology vol.(35), 1982, pp. 187–192.

[4] L. Breiman, "Bagging predictors," Machine Learning vol.(24), no. (2), 1996, pp.123–140.

[5] S. Diaz, D. Nelson, G. Deferrari, and C. Camilion, "Estimated and measured DNA, plant-chromosphere and erythemal-weighted irradiances at Barrow and South Pole (1979–2000)," Agricultural and Forest Meteorology, vol.(120), 2003, pp.69–82.

[6] S. Diaz, D. Nelson, G. Deferrari, and C. Camilion, "A model to extend spectral and multi-wavelength UV irradiances time series: model development and validation," Journal of Geophysical Research vol. (108), no. (D4), 2003, p.4150

[7] H. K. Elminir, "Dependence of urban air pollutants on meteorology," Science of The Total Environment, vol. (350), no.(1-3), 2005, pp.225 – 237.

[8] H. K. Elminir, H. S. Own, Y. A. Azam, A.M. Riad and F. I. Younis, "Testing the applicability of artificial intelligence techniques to the subject of erythemal ultraviolet solar radiation. Part two An intelligent system based on multi-classifier technique," Journal of Photochemistry and Photobiology B: Biology vol. (90), 2008, pp. 198–206.

[9] U. Feister, J. Junk, "Re-construction of daily solar UV irradiation by an artificial neural network (ANN), "in: Proceedings of SPIE – The International Society for Optical Engineering, 2006, vol. 6362, Article No. 63622H.

[10] V. Fioletov, G. Bodeker, A. Miller, R. McPeters, and R. Stolarski, "Global and zonal total ozone variations estimated from ground-based and satellite measurements: 1964–2000," Journal of Geophysical Research vol. (107), no. (D22), 2002, p. 4647, doi:10.1029/2001JD001350.

[11] L. Gantner, P. Winkler, and U. Kooehler, "A method to derive long-term time series and trends of UV-B radiation (1968–997) from observations at Hohenpeissenberg (Bavaria)," Journal of Geophysical Research, vol. (105), no. (D4), 2000, pp. 4879–4888.

[12] G. Helena, Using multiple classifiers in statistical pattern recognition. GE Global Research, Report No. 2003GRC375, January 2004.

[13] J. Herman, P. Bhartia, J. Ziemke, Z. Ahmad, and D. Larko, "UV-B increases (1979–1992) from decreases in total ozone," Journal of Geophysical Research Letter vol. (23), no. (16), 1996, pp. 2117–2120.

[14] T. Hothorn, and B. Lausen, "Bundling classifiers by bagging trees," Computational Statistics and Data Analysis vol. (49), 2005, pp. 1068–1078.

[15] I. H. Witten, and E. Frank, "Data Mining: Practical machine learning tools and techniques," second ed., Morgan Kaufmann, June 2005. ISBN: 0-12-088407-0, Elsevier LTD, Oxford.

[16] J. R. Jang, C. T. Sun, and E. Mizutani, "Neuro-fuzzy and Soft Computing," Computational Approach to Learning and Machine Intelligence. Prentice Hall Publishing Co., 1997. ISBN 0-471-16003-2, p. 607.

[17] Z. Jianpei, C. Lili, and M. Jun, "A new multiple classi.ers combination algorithm," IEEE First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 20–24 April 2006, vol. 2, pp. 287–291.

[18] S. A. Kalogirou, "Applications of artificial neural networks in energy systems: A review," Energy Conversion and Management vol. (40), 1999, pp.1073–1087.

[19] J. Kaurola, P. Taalas, T. Koskela, J. Borkowski, and W. Josefsson, "Long term variations of UV-B doses at three stations in northern Europe," Journal of Geophysical Research, vol. (105), no. (D16), 2000, pp. 813–820.

[20] A. Kazantzidis, A.F. Bais, D.S. Balis, E. Kosmidis, and C.S. Zerefos, "Sensitivity of solar UV radiation to ozone and temperature pro.les at Thessaloniki (40.5_ N, 23_ E), Greece," Journal of Atmospheric and Solar-Terrestrial Physics vol. (67), 2005, pp. 1321–1330.

[21] A. Kylling, and B. Mayer, "Ultraviolet radiation in partly snow covered terrain: observations and three-dimensional simulations," Geophysical Research Letters vol. (28), 2001, pp. 3665–3668.

[22] A. Lindfors, A. Arola, J. Kaurola, and P. Taalas, "Long-term erythemal UV doses at Sodankyla¨ estimated using total ozone, sunshine duration, and snow depth," Journal of Geophysical Research vol. (108), no. (D16), 2003, pp. 4518–4529, doi:10.1029/2002JD003325.

[23] B. Mayer, C.A. Fischer, and S. Madronich, "Estimation of surface actinic flux from satellite (TOMS) ozone and cloud reflectivity measurements," Geophysical Research Letters vol. (25), 1998, pp. 4321–4324.

[24] T. Henriksen, A. Dahlback, S. Larsen, and J. Moan, "Ultraviolet radiation and skin cancer: Effect of ozone layer depletion," Journal of Photochemistry and Photobiology, vol. 51,1990, pp. 579–582.

[25] W. Murillo, J. Cañada, and G. Pedrós, "Correlation between ultraviolet (290–385 nm) and global irradiation in Valencia and Cordoba (Spain)," Renewable Energy, vol. (28), 2003, pp. 409 – 418.

[26] S. M. Robaa, "A study of ultraviolet solar radiation at Cairo urban area, Egypt, "Solar Energy vol. (77), 2004, pp. 251–259.

[27] T. Rohlfing, and C.R. Maurer, "Multi-classifier framework for atlas-based image segmentation," Pattern Recognition Letters vol. (26), 2005, pp.2070–2079.

[28] A. Ruggaber, R. Dlugi, and T. Nakajima, "Modeling radiation quantities and photolysis frequencies in the troposphere," Journal of Atmospheric Chemistry vol. (18), 1994, pp. 171–210.

[29] Y. Sazi Murat, "Comparison of fuzzy logic and artificial neural networks approaches in vehicle delay modeling," Transportation Research Part C 14, 2006, pp. 316–334.

[30] P. Weihs, and A.R. Webb, Accuracy of spectral UV model calculations, 1, consideration of uncertainties in input parameters, Journal of Geophysical Research, vol. 102, 1997, pp. 1541–1550.

[31] J.R. Ziemke, S. Chandra, J. Herman, and C. Varotsos, "Erythemally weighted UV trends over northern latitudes derived from Nimbus 7 TOMS measurements," Journal of Geophysical Research 105 (D6) (2000) 7373–7382.

[32] H. M. El-Bakry, "New Faster Normalized Neural Networks for Sub-Matrix Detection using Cross Correlation in the Frequency Domain and Matrix Decomposition," Applied Soft Computing journal, vol. 8, issue 2, March 2008, pp. 1131-1149.

[33] H. M. El-Bakry and M. Hamada, "A New Implementation for High Speed Neural Networks in Frequency Space," Lecture Notes in Computer Science, Springer, KES 2008, Part I, LNAI 5177, pp. 33-40.

[34] H. M. El-Bakry, and N. Mastorakis "New Fast Normalized Neural Networks for Pattern Detection,"

Image and Vision Computing Journal, vol. 25, issue 11, 2007, pp. 1767-1784.

[35] H. M. El-Bakry and N. Mastorakis, "Fast Code Detection Using High Speed Time Delay Neural Networks," Lecture Notes in Computer Science, Springer, vol. 4493, Part III, May 2007, pp. 764-773.

[36] H. M. El-Bakry, "New Fast Principal Component Analysis for Face Detection," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol.11, no.2, 2007, pp. 195-201

[37] H. M. El-Bakry, and Q. Zhao, "A Modified Cross Correlation in the Frequency Domain for Fast Pattern Detection Using Neural Networks," International Journal of Signal Processing, vol.1, no.3, pp. 188-194, 2004.

[38] H. M. El-Bakry, and Q. Zhao, "Fast Object/Face Detection Using Neural Networks and Fast Fourier Transform," International Journal of Signal Processing, vol.1, no.3, pp. 182-187, 2004.

[39] H. M. El-Bakry, and Q. Zhao, "Fast Pattern Detection Using Normalized Neural Networks and Cross Correlation in the Frequency Domain," EURASIP Journal on Applied Signal Processing, Special Issue on Advances in Intelligent Vision Systems: Methods and Applications—Part I, Vol. 2005, No. 13, 1 August 2005, pp. 2054-2060.

[40] H. M. El-Bakry, and Q. Zhao, "A Fast Neural Algorithm for Serial Code Detection in a Stream of Sequential Data," International Journal of Information Technology, vol.2, no.1, pp. 71-90, 2005.

[41] H. M. El-Bakry, and H. Stoyan**,** "FNNs for Code Detection in Sequential Data Using Neural Networks for Communication Applications, " Proc. of the First International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2004, 21-25 July, 2004. Orlando, Florida, USA, Vol. IV, pp. 150-153.

[42] H. M. El-Bakry, "Face detection using fast neural networks and image decomposition," Neurocomputing Journal, vol. 48, 2002, pp. 1039-1046.

[43] H. M. El-Bakry, "Human Iris Detection Using Fast Cooperative Modular Neural Nets and Image Decomposition," Machine Graphics & Vision Journal (MG&V), vol. 11, no. 4, 2002, pp. 498-512.

[44] H. M. El-Bakry, "Automatic Human Face Recognition Using Modular Neural Networks," Machine Graphics & Vision Journal (MG&V), vol. 10, no. 1, 2001, pp. 47-73.

[45] H. M. El-Bakry, and Q. Zhao, "Fast Pattern Detection Using Neural Networks Realized in Frequency Domain," Proc. of the International Conference on Pattern Recognition and Computer Vision, The Second World Enformatika Congress WEC'05, Istanbul, Turkey, 25-27 Feb., 2005, pp. 89-92.

[46] H. M. El-Bakry, and Q. Zhao, "Sub-Image Detection Using Fast Neural Processors and Image Decomposition," Proc. of the International Conference on Pattern Recognition and Computer Vision, The Second World Enformatika Congress WEC'05, Istanbul, Turkey, 25-27 Feb., 2005, pp. 85-88.

[47]H. M. El-Bakry, and Q. Zhao, "Fast Normalized Neural Processors For Pattern Detection Based on Cross Correlation Implemented in the Frequency Domain," Journal of Research and Practice in Information Technology, Vol. 38, No.2, May 2006, pp. 151-170.

[48] H. M. El-Bakry, and Q. Zhao, "New Faster Normalized Neural Networks For Sub-Matrix Detection Using Cross Correlation in the Frequency Domain and Matrix Decomposition," International Journal of Signal Processing, vol.2, no.3, 2005, pp. 183-202.

[49] H. M. El-Bakry, and Q. Zhao, "Speeding-up Normalized Neural Networks For Face/Object Detection," Machine Graphics & Vision Journal (MG&V), vol. 14, No.1, 2005, pp. 29-59.

[50] H. M. El-Bakry, and Q. Zhao, "Fast Time Delay Neural Networks," the International Journal of Neural Systems, vol. 15, No.6, December 2005, pp.445-455.

[51] H. M. El-Bakry, "A New Implementation of PCA for Fast Face Detection," International Journal of Intelligent Technology, Vol. 1, No.2, 2006, pp. 145-153.

[52] H. M. El-Bakry, "New Fast Time Delay Neural Networks Using Cross Correlation Performed in the Frequency Domain," Neurocomputing Journal, vol. 69, October 2006, pp. 2360-2363.

[53] H. M. El-bakry, and Q. Zhao, "Modified Time Delay Neural Networks For Fast Data Processing," Proc. of IEEE Eighth International Symposium on Signal Processing and its Applications, Sydney, Australia, August 28-31, 2005.

[54] H. M. El-Bakry, "New High Speed Normalized Neural Networks for Fast Pattern Discovery on Web Pages," the International Journal of Computer Science and Network Security, vol.6, No. 2A, Feb. 2006, pp.142-152.

[55] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. 19, 1965, pp. 297–301.

[56] R. Klette, and Zamperon, "Handbook of image processing operators, " John Wiley & Sonsltd, 1996.

[57] A. Hirose, "Complex-Valued Neural Networks Theories and Applications", Series on innovative Intelligence, vol.5. Nov. 2003.

[58] http://toms.gsfc.nasa.gov.

[59] R. D. McPeters, P. K. Bhartia, A. J. Krueger, and J. R. Herman, "Nimbus–7 Total Ozone Mapping Spectrometer (TOMS) Data Products User's Guide", NASA, Reference Publication 1996.

[60] U. Feister, E. Jakel, and K. Gericke, "Parameterization of daily solar global ultraviolet irradiation," Journal of Photochemistry and Photobiology vol. (76), no. (3), 2002,pp. 281–293.

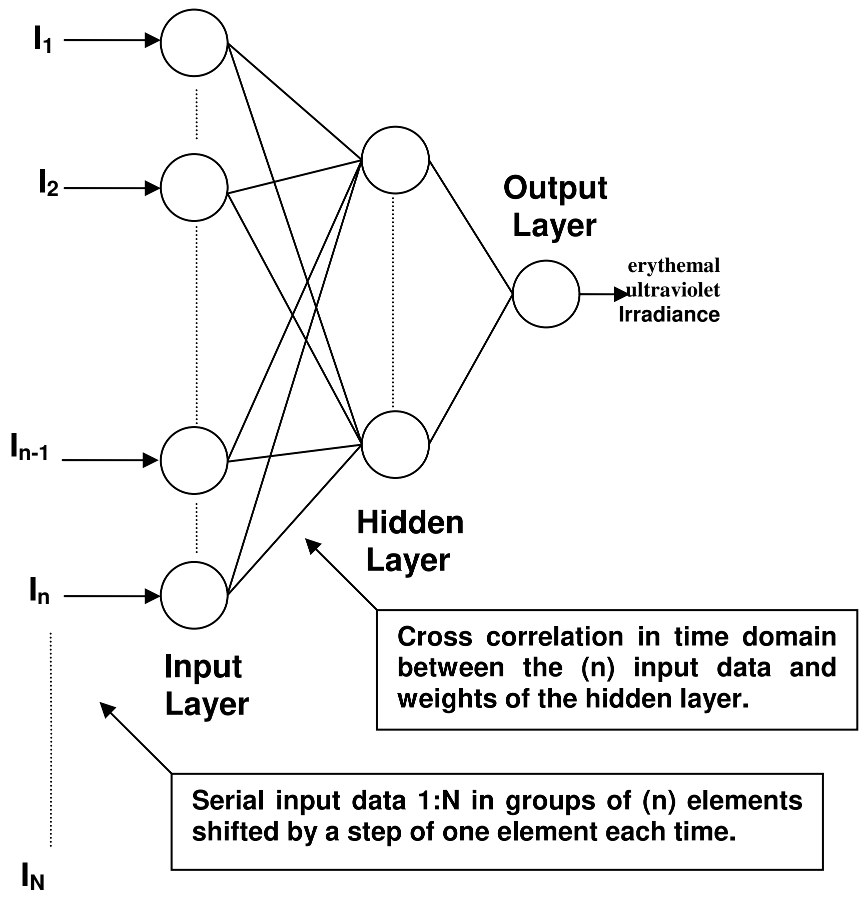
I1
I2
In-1
In
IN
Input Layer
Hidden Layer
Output Layer
erythemal ultraviolet Irradiance
Cross correlation in time domain between the (n) input data and weights of the hidden layer.
Serial input data 1:N in groups of (n) elements shifted by a step of one element each time.

Fig. 1. Classical Feed-forward neural networks.

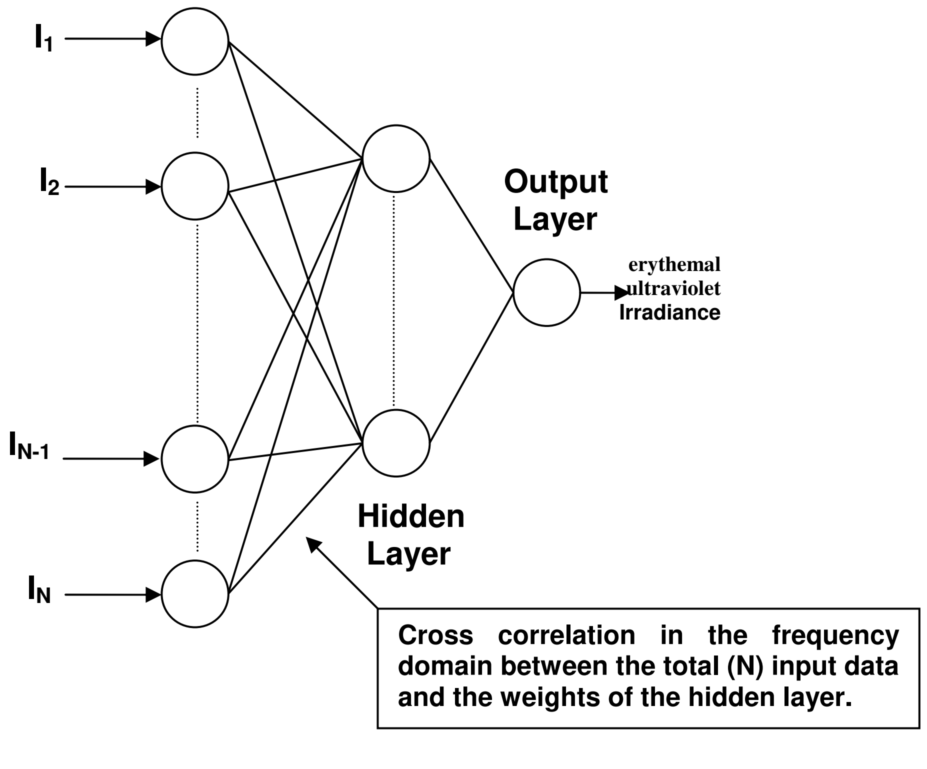
I1
I2
IN-1
IN
Hidden Layer
Output Layer
erythemal ultraviolet Irradiance
Cross correlation in the frequency domain between the total (N) input data and the weights of the hidden layer.

Fig.2. Fast neural networks.

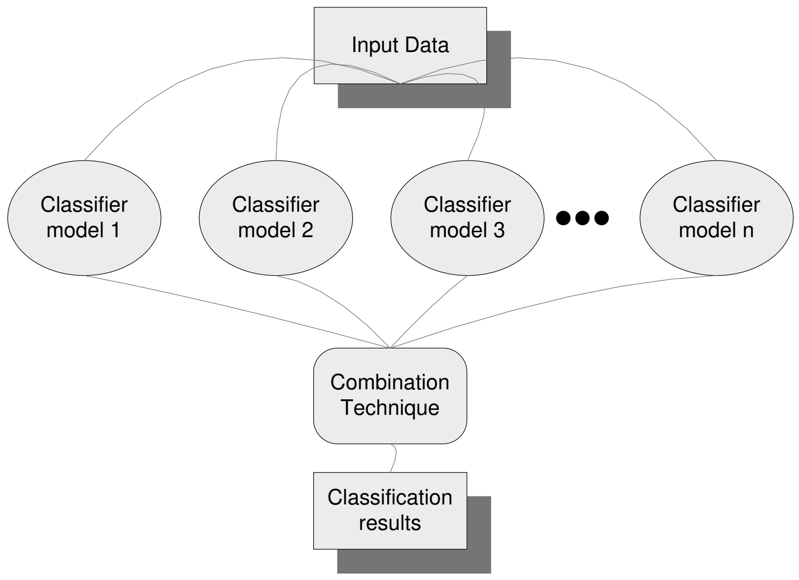

Fig. 3. Information Fusion by using combined classifiers.

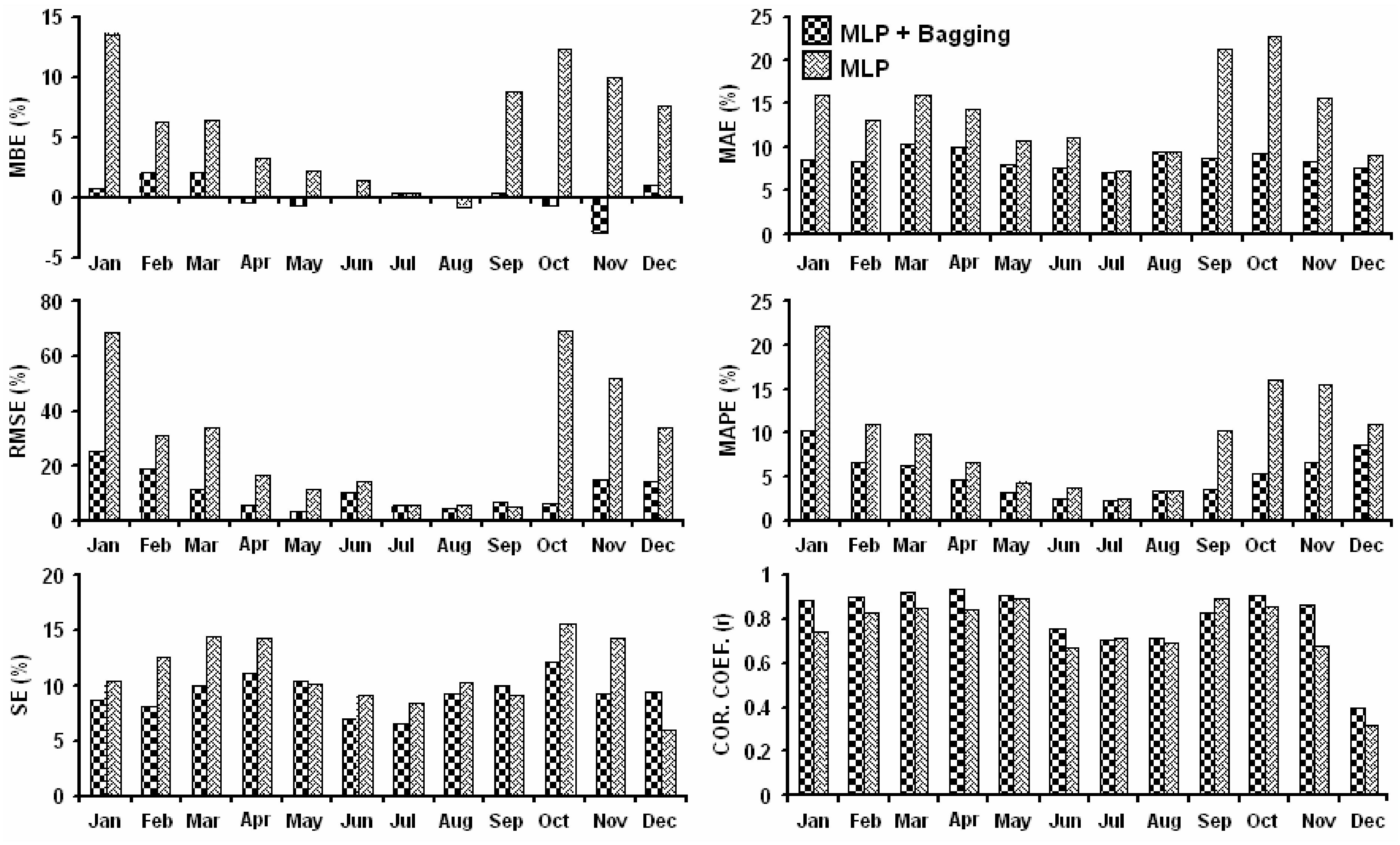

Fig. 4. Error values associated with erythemal ultraviolet irradiance estimates from MLP based *bagging* and MLP models at Cairo site.

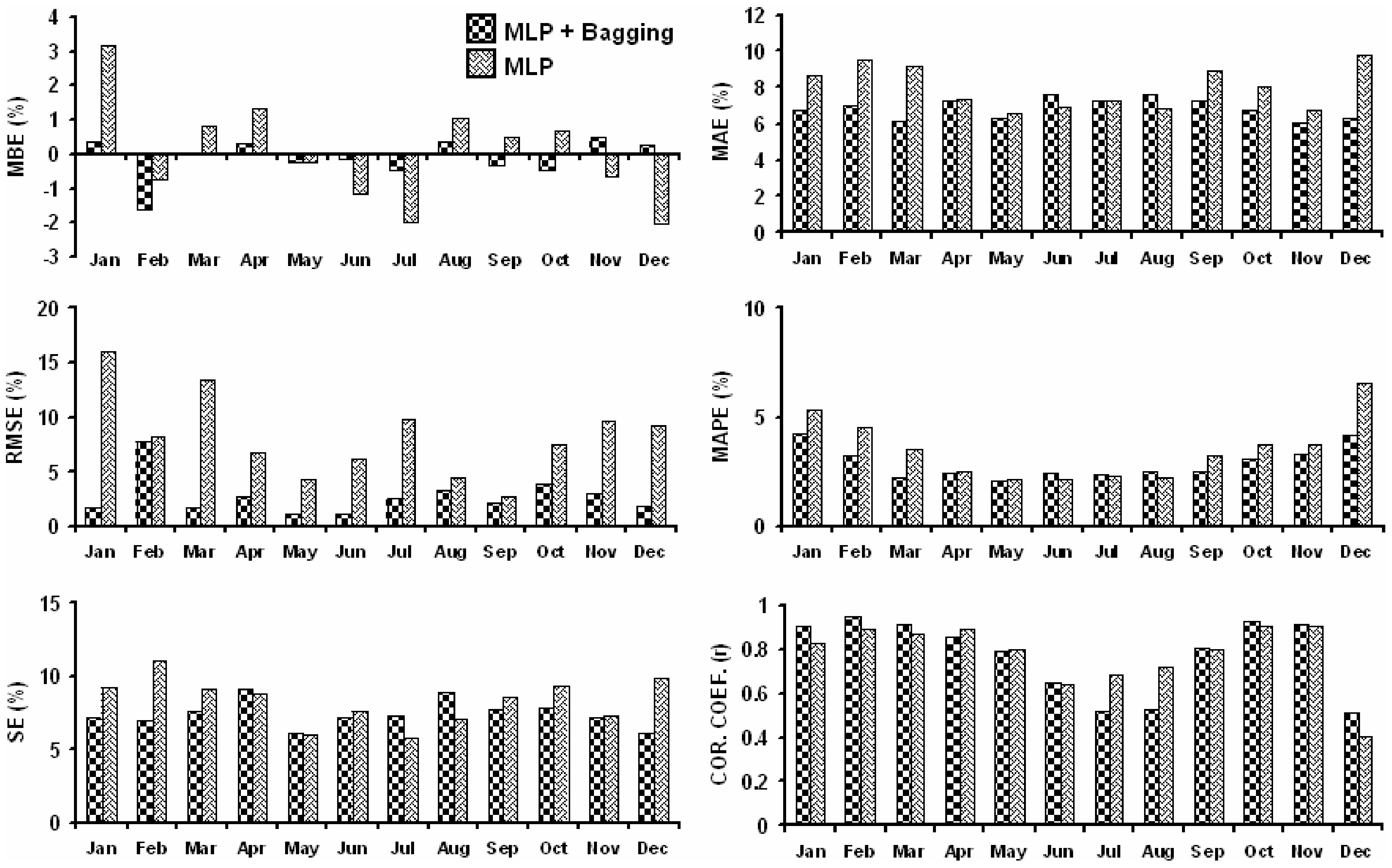

Fig. 5. Error values associated with erythemal ultraviolet irradiance estimates from MLP based bagging and MLP models at Aswan site.

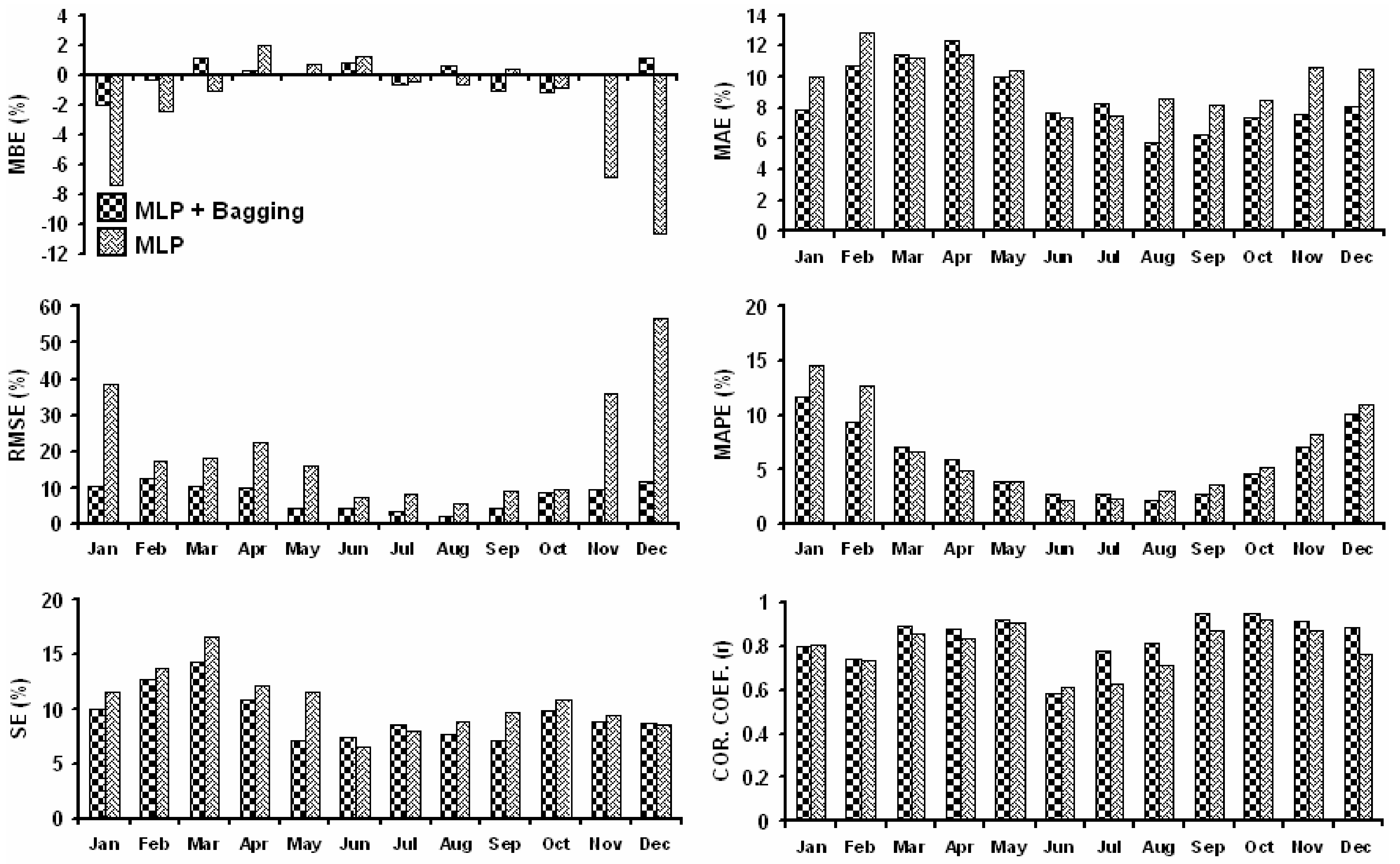

Fig. 6. Error values associated with erythemal ultraviolet irradiance estimates from MLP based bagging and MLP models at Mersa-Matruh site.

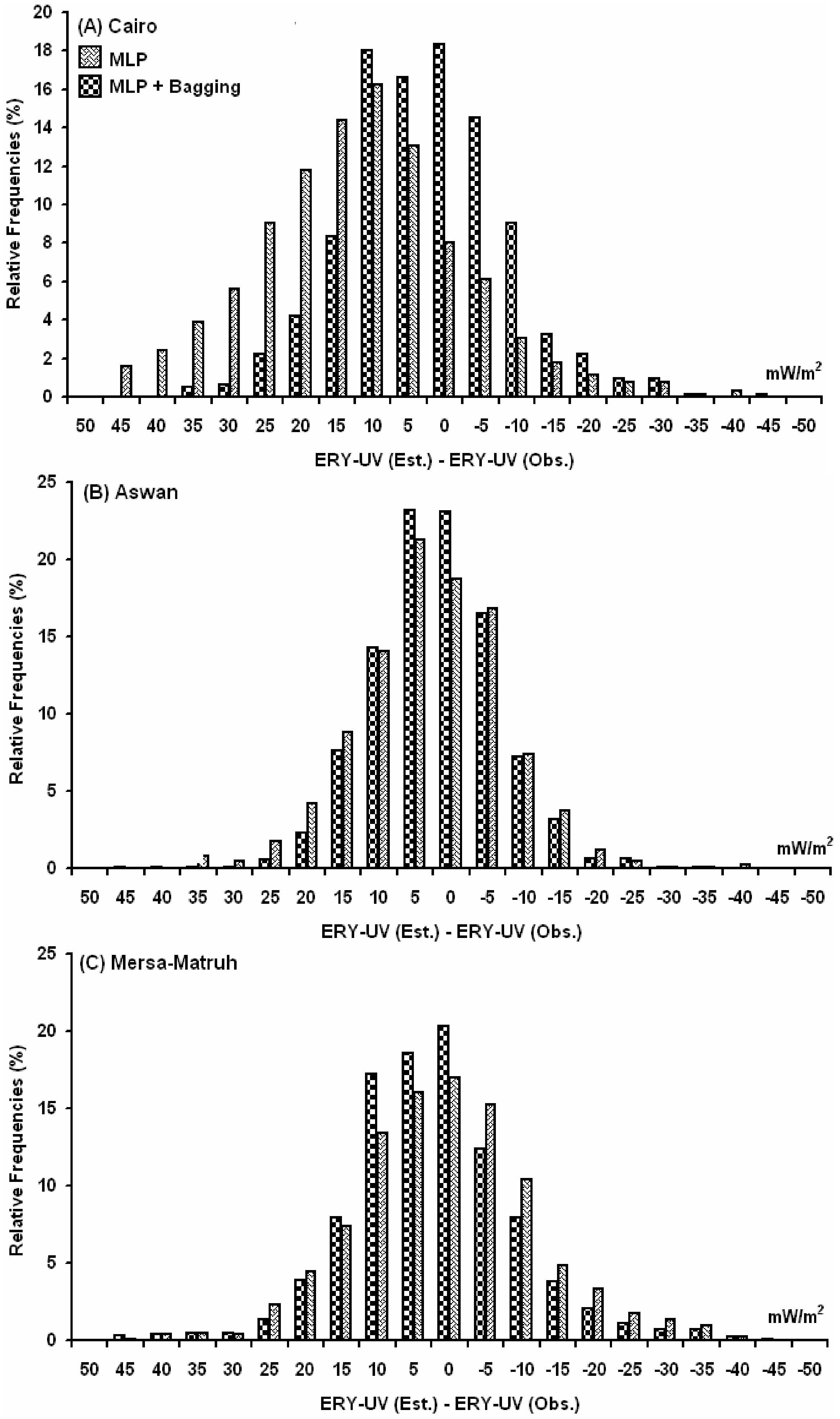

Fig. 7. The frequency distribution of the differences between predicted and observed values.

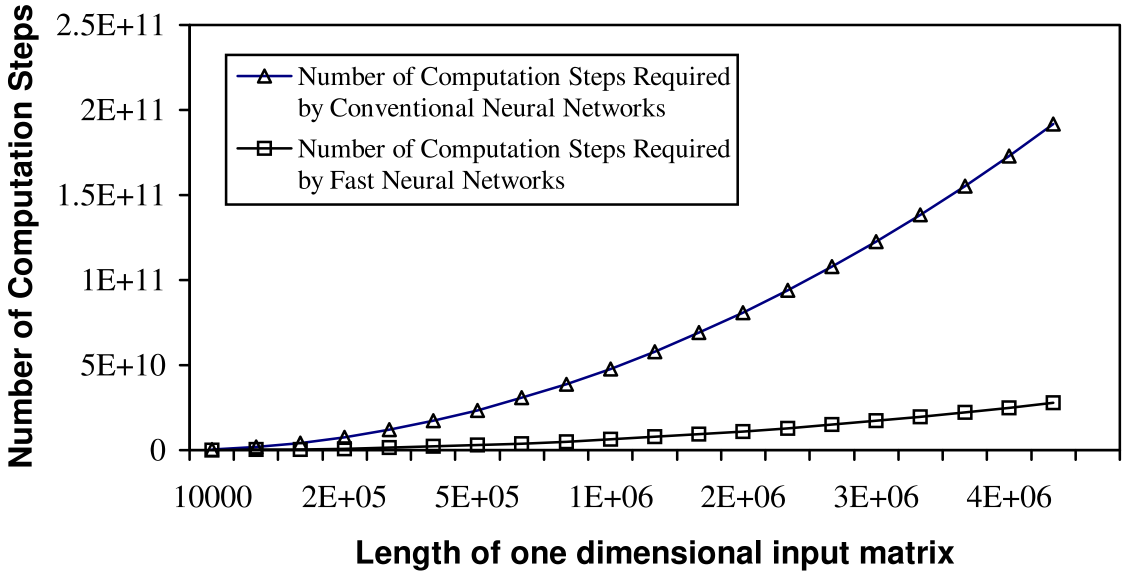

Fig. 8. A comparison between the number of computation steps required by FNNS and CNNs in case of real-valued one dimensional input matrix and complex-valued weight matrix (n=400).

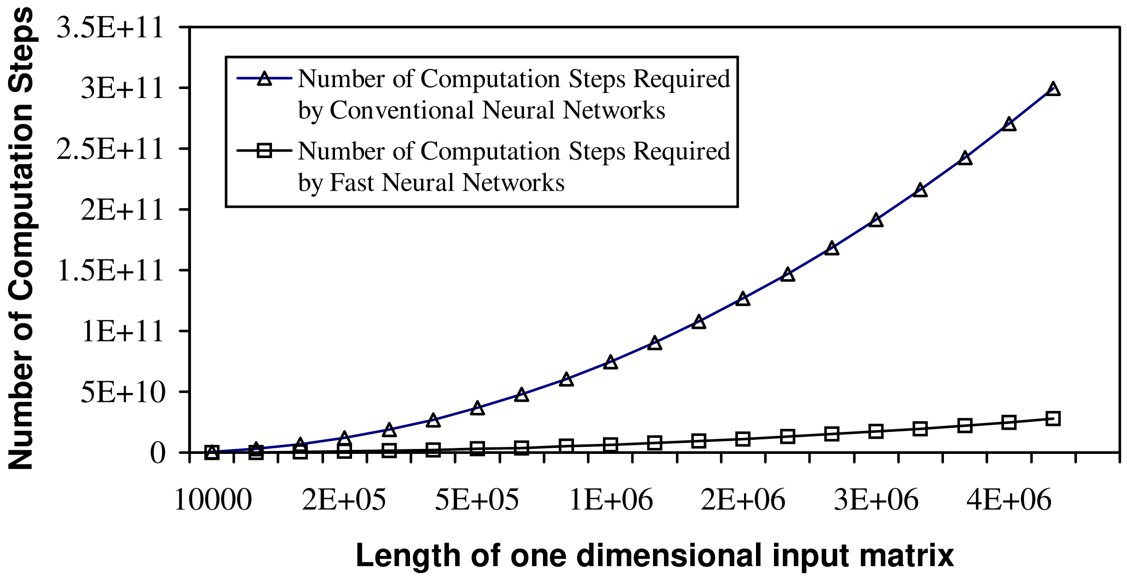

Fig. 9. A comparison between the number of computation steps required by FNNS and CNNs in the case of real-valued one dimensional input matrix and complex-valued weight matrix (n=625).

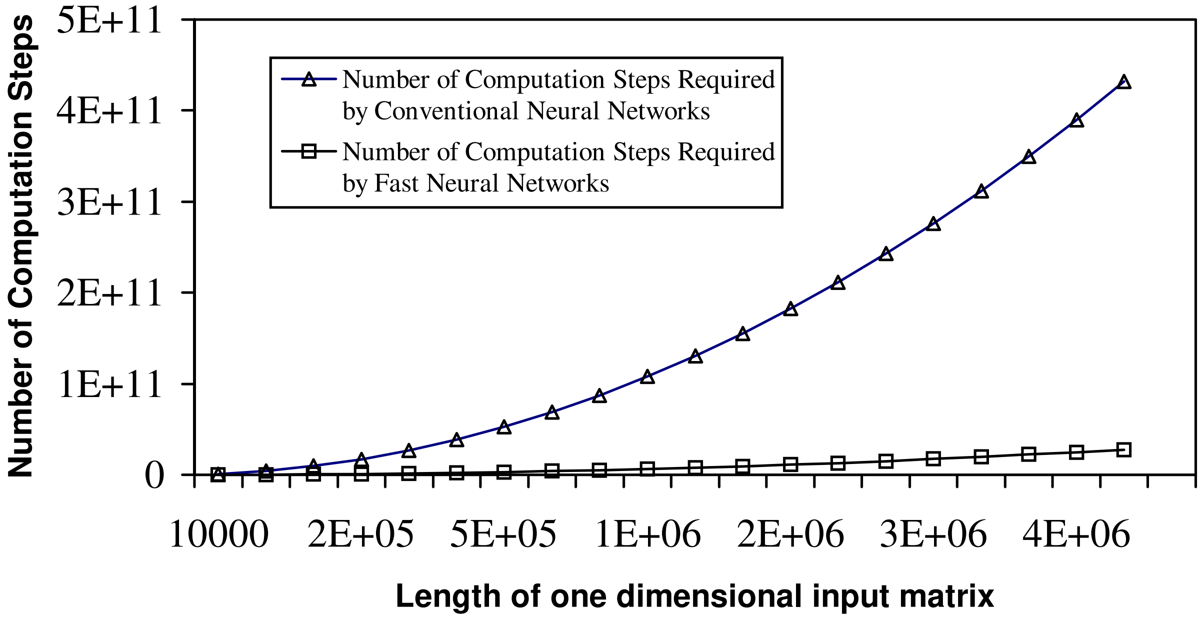

Fig. 10. A comparison between the number of computation steps required by FNNS and CNNs in the case of real-valued one dimensional input matrix and complex-valued weight matrix (n=900).

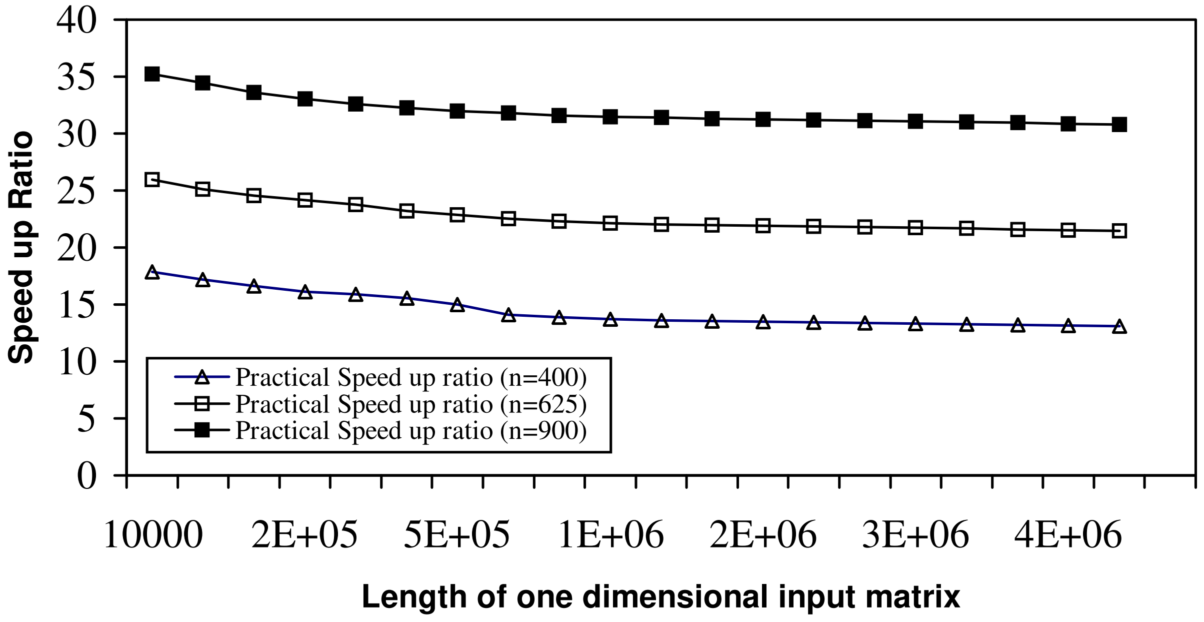

Fig. 11. Practical speed up ratio for FNNs in case of one dimensional real-valued input matrix and complex-valued weights.

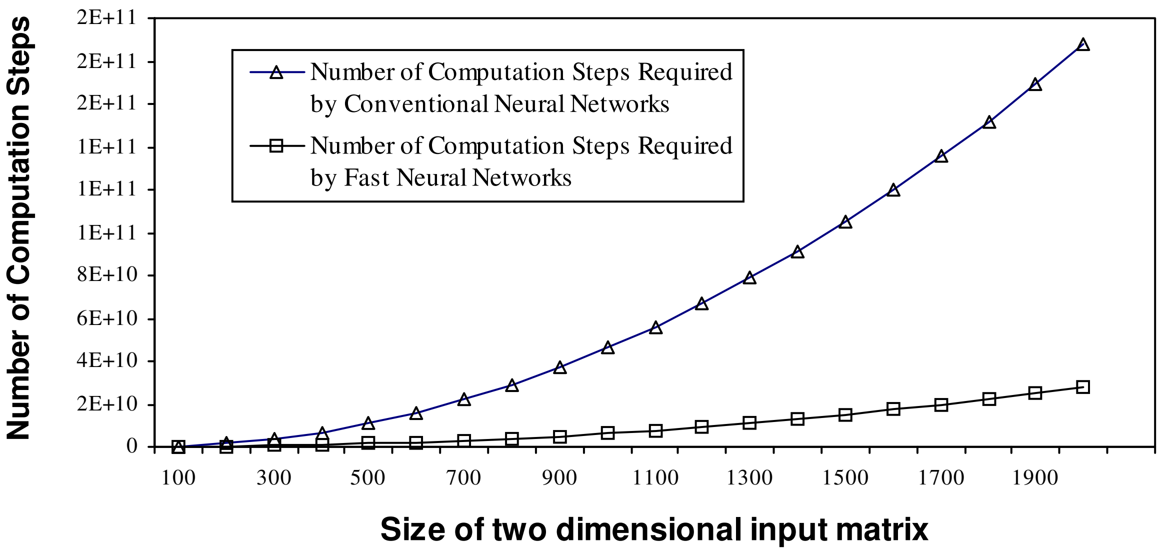

Fig. 12. A comparison between the number of computation steps required by FNNS and CNNs in the case of real-valued two dimensional input matrix and complex-valued weight matrix (n=20).

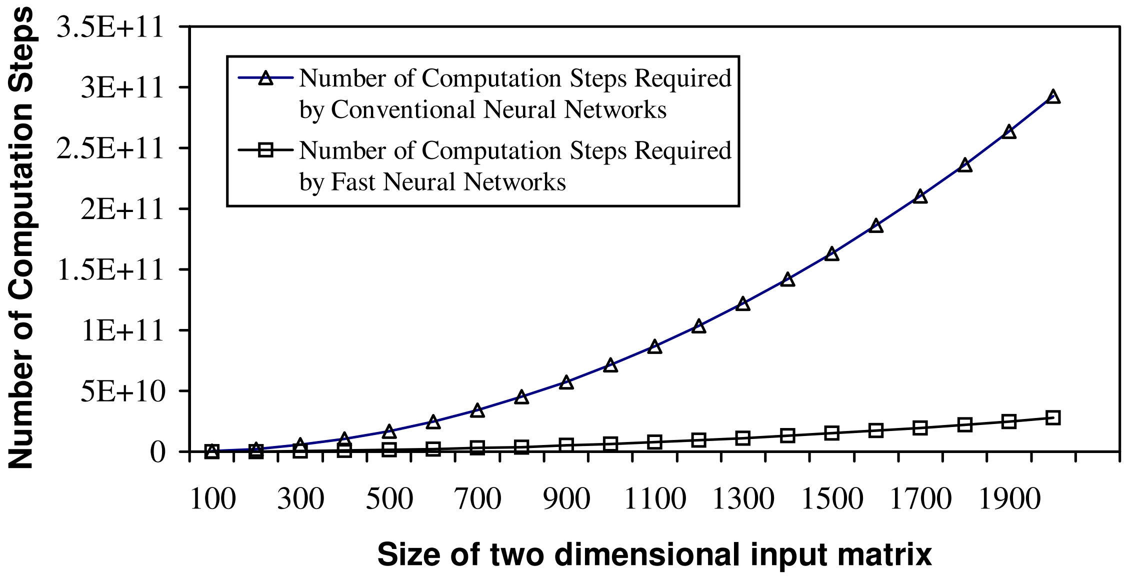

Fig. 13. A comparison between the number of computation steps required by FNNS and CNNs in the case of real-valued two dimensional input matrix and complex-valued weight matrix (n=25).

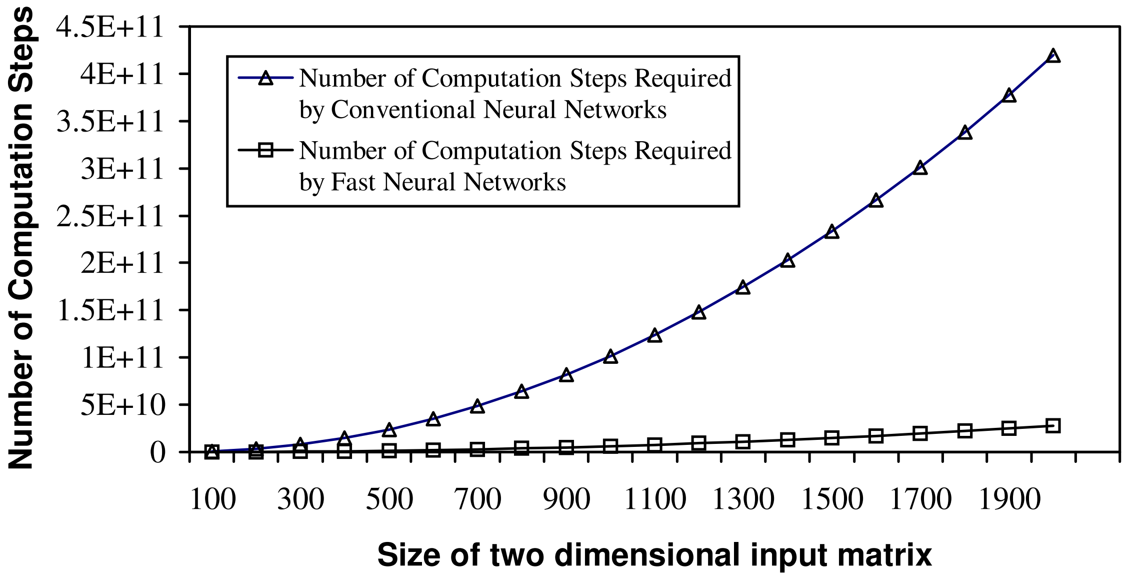

Fig. 14. A comparison between the number of computation steps required by FNNS and CNNs in the case of real-valued two dimensional input matrix and complex-valued weight matrix (n=30).

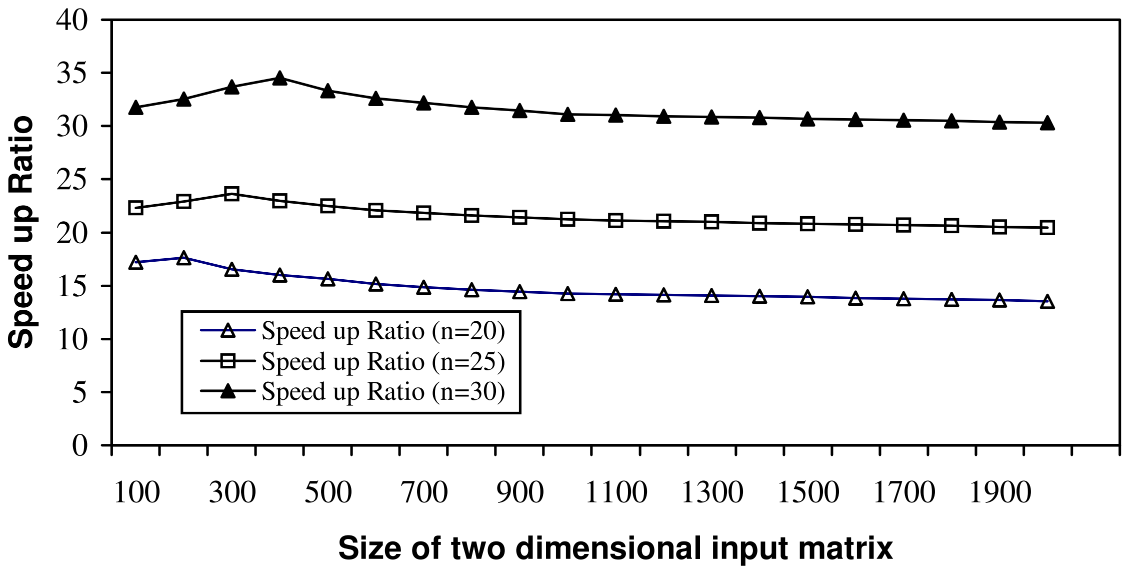

Fig. 15. Practical speed up ratio for FNNs in case of two dimensional real-valued input matrix and complex-valued weights.

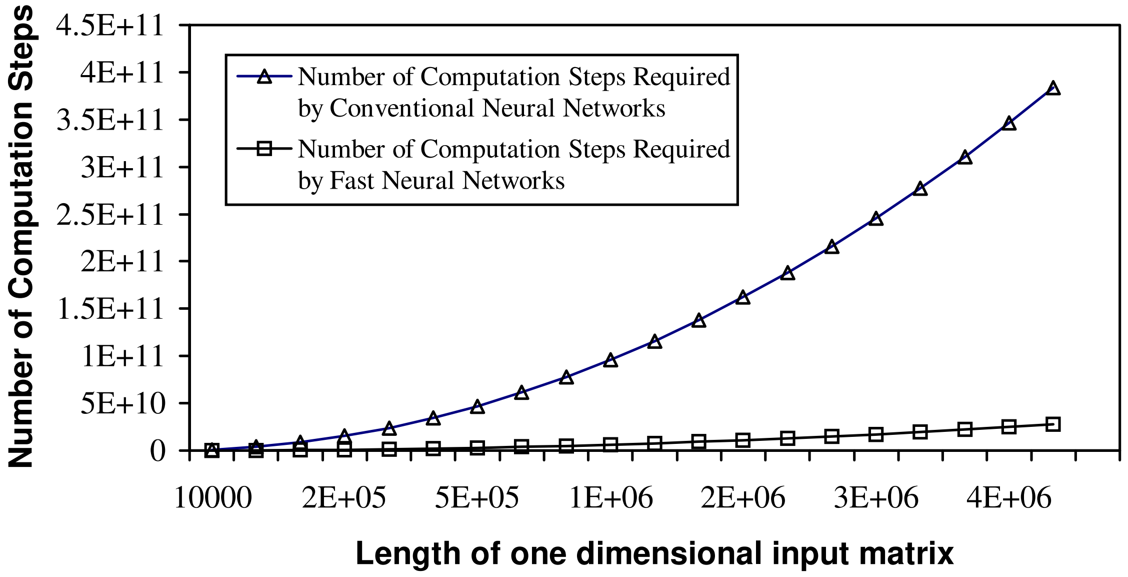

Fig. 16. A comparison between the number of computation steps required by FNNS and CNNs in the case of complex-valued one dimensional input matrix and complex-valued weight matrix (n=400).

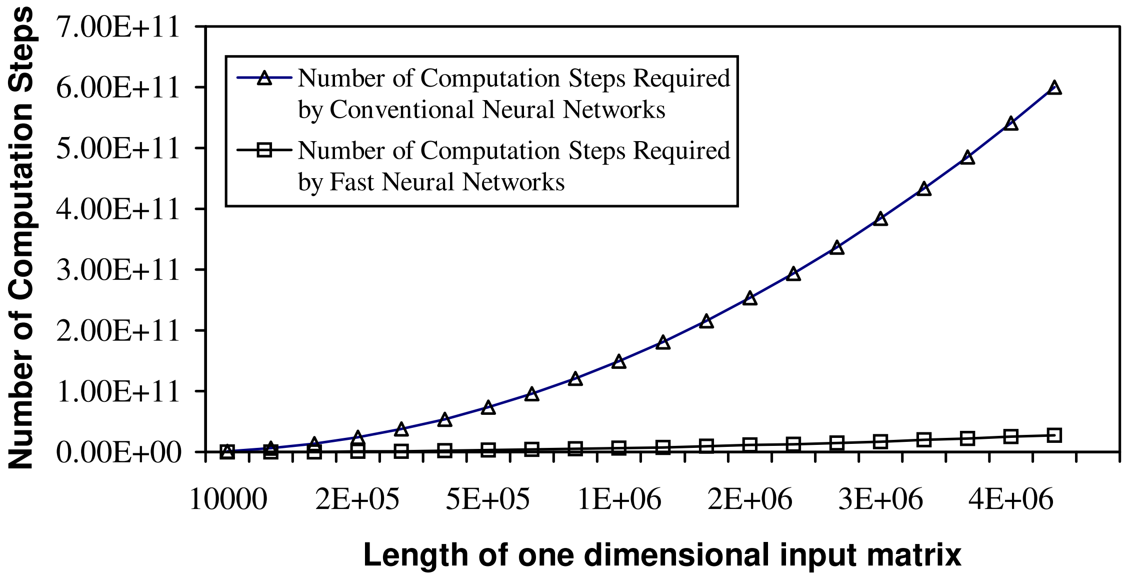

Fig. 17. A comparison between the number of computation steps required by FNNS and CNNs in the case of complex-valued one dimensional input matrix and complex-valued weight matrix (n=625).

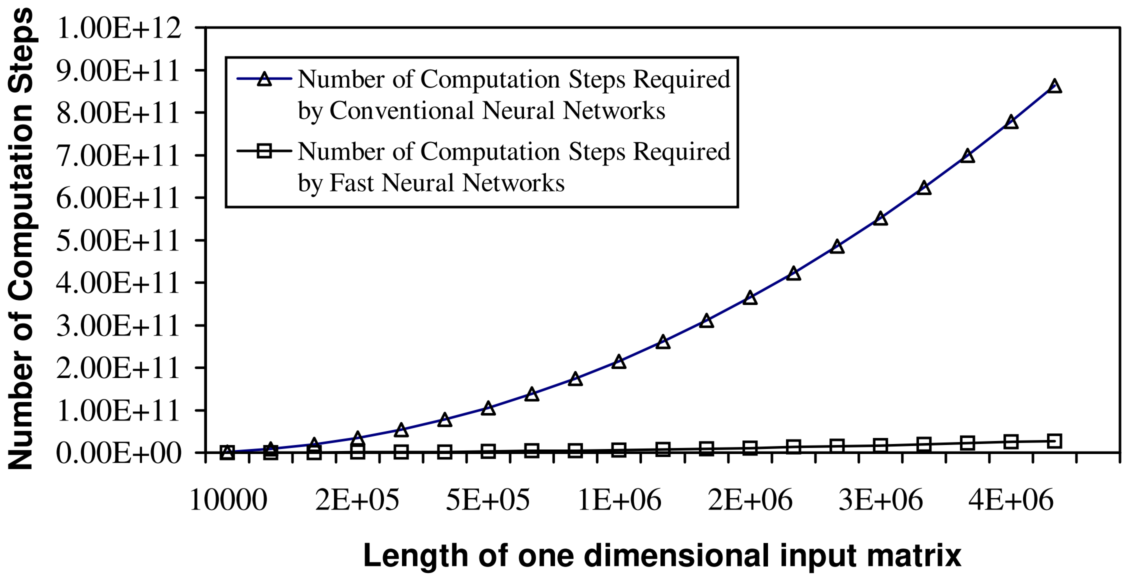

Fig. 18. A comparison between the number of computation steps required by FNNS and CNNs in the case of complex-valued one dimensional input matrix and complex-valued weight matrix (n=900).

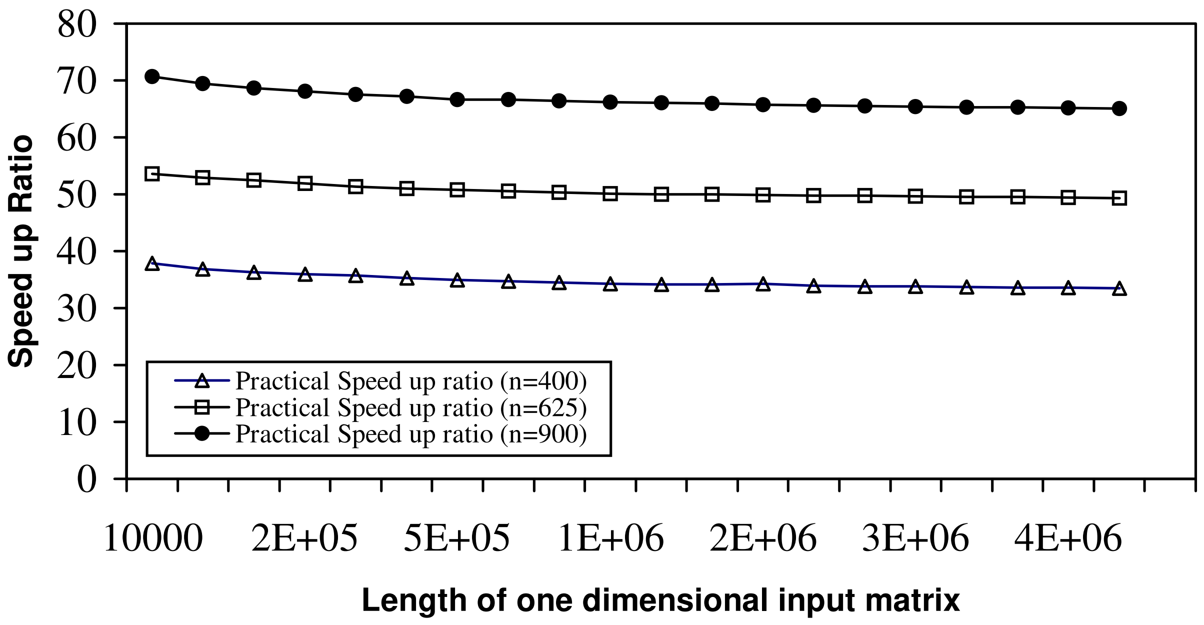

Fig. 19. Practical speed up ratio for FNNs in case of one dimensional complex-valued input matrix and complex-valued weights.

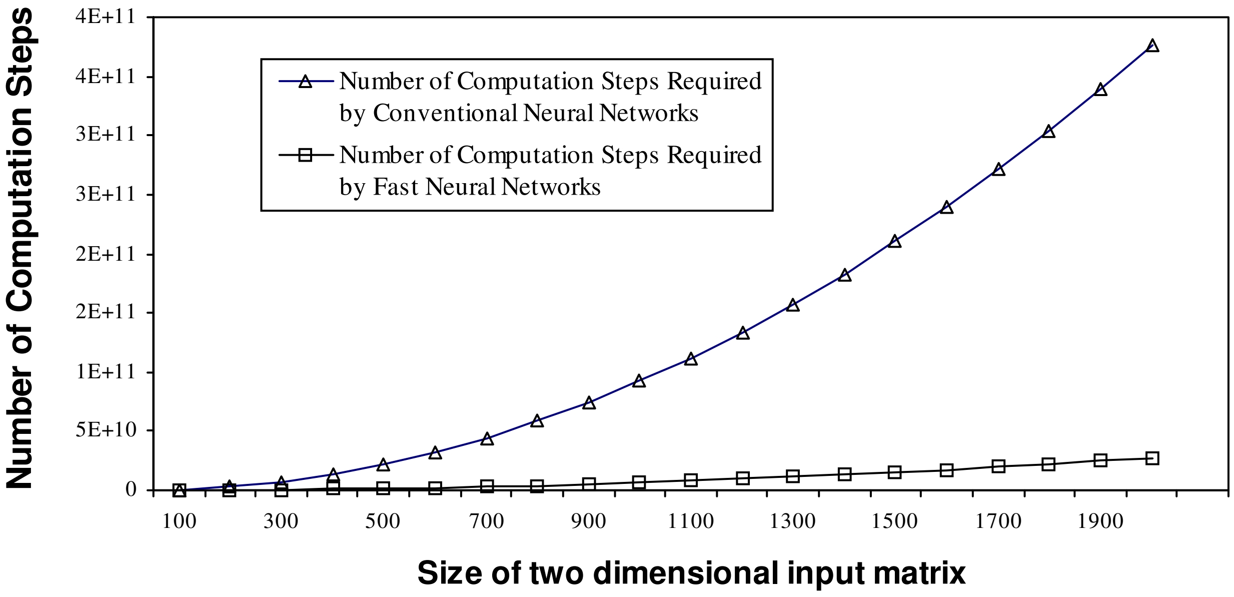

Fig. 20. A comparison between the number of computation steps required by FNNS and CNNs in the case of complex-valued two dimensional input matrix and complex-valued weight matrix (n=20).

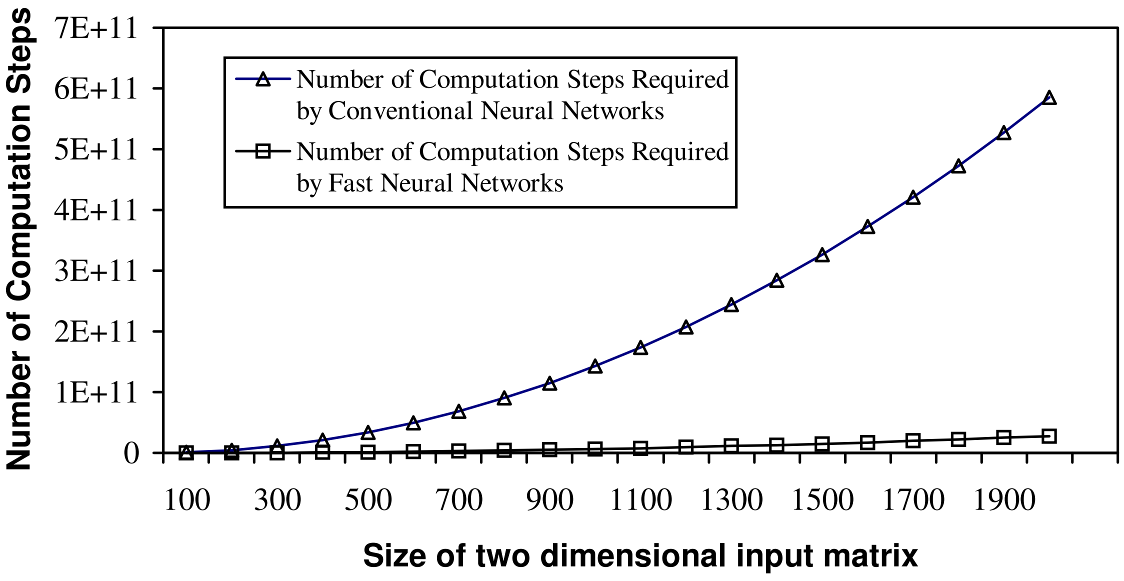

Fig. 21. A comparison between the number of computation steps required by FNNS and CNNs in the case of complex-valued two dimensional input matrix and complex-valued weight matrix (n=25).

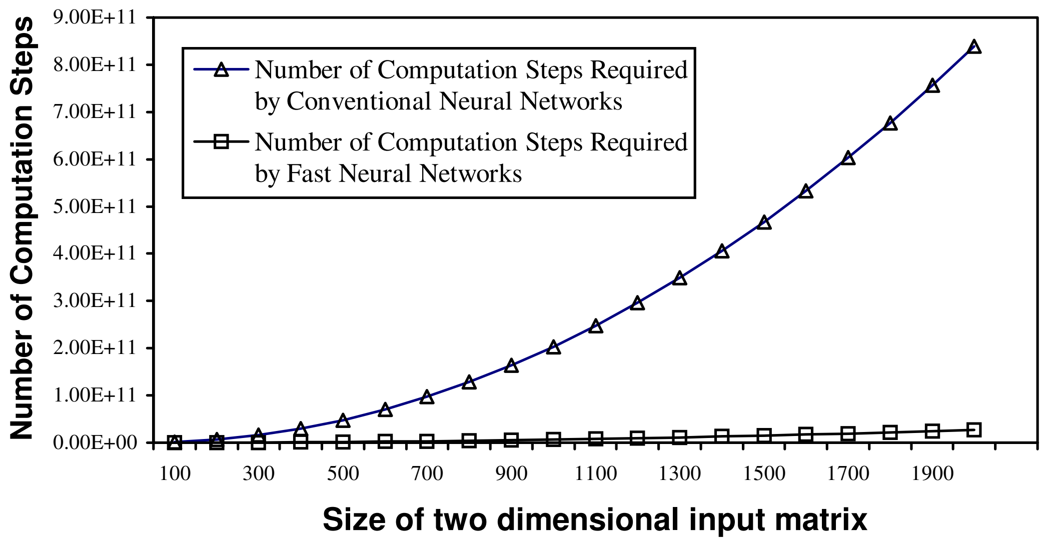

Fig. 22. A comparison between the number of computation steps required by FNNS and CNNs in the case of complex-valued two dimensional input matrix and complex-valued weight matrix (n=30).

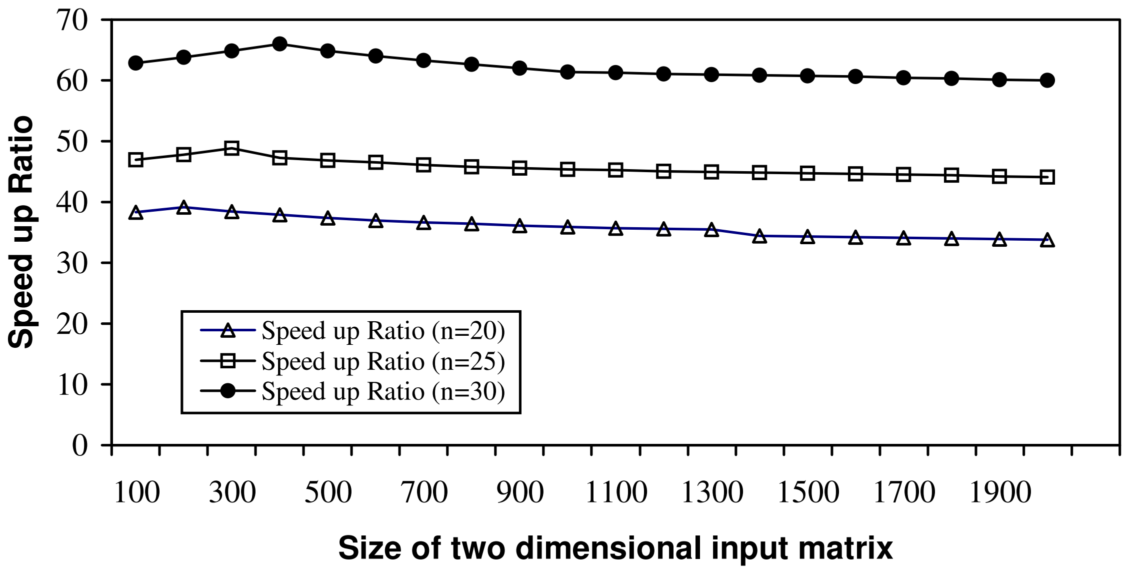

Fig. 23. Practical speed up ratio for FNNs in case of two dimensional complex-valued input matrix in and complex-valued weights.