METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY)

HAIDER KHALAF JABBAR
University of Misan, Misan, Iraq and, Department of Computer Science
Aligarh Muslim University, Aligarh-202002 (U.P), INDIA
haider.allamy@gmail.com, he25_allamy@yahoo.com

DR. RAFIQUL ZAMAN KHAN
Associate Professor, Department of Computer Science,
Aligarh Muslim University, Aligarh-202002 (U.P), INDIA
rzk32@yahoo.co.in. rzkhan.cs@mail.amu.ac.in

ABSTRACT: Machine learning is an important task for learning artificial neural networks, and we find in the learning one of the common problems of learning the Artificial Neural Network (ANN) is over-fitting and under-fitting to outlier points. In this paper we performed various methods in avoiding over-fitting and under-fitting; that is penalty and early stopping methods. A comparative study has been presented for the aforementioned methods to evaluate their performance within a range of specific parameters such as; speed of training, over-fitting and under-fitting avoidance, difficulty, capacity, time of training, and their accuracy. Besides these parameters we have included comparison between over-fitting and under-fitting. We found the early stopping method as being better as compared to the penalty method, as it can avoid over-fitting and under-fitting with respect to validation time. Besides we find that Under-fitting neural networks perform poorly on both training and test sets, but Over-fitting networks may do very well on training sets though terribly on test sets.

KEYWORDS: Machine learning, over-fitting, under-fitting, early stopping, penalty method

INTRODUCTION

One of the common problems of the use of ANN is the over-fitting to outlier points (Wang, J.H.; Jiang, J.H.; Yu, R.Q. 1996). Over-fitting is a key problem in the supervised machine learning tasks. It is the phenomenon detected when a learning algorithm fits the training data set so well that noise and the peculiarities of the training data are memorized. According to the result of learning algorithms performance drops when it is tested in an unknown data set. The amount of data used for learning process is fundamental in this context. Small data sets are more prone to over-fitting than large data sets, and despite the complexity of some learning problem, large data sets can even be affected by over-fitting (Santos, E. M., Sabourin, R., & Maupin, P. 2009). Over-fitting of the training data leads to deterioration of generalization properties of the model, and results in its untrustworthy performance when applied to novel measurements. Hence the purpose of the methods to avoid over-fitting is somehow contradictory to the goal of optimization algorithms, which aims at finding the best possible solution in parameter space according to pre-defined objective function and available data. Moreover, different optimization algorithms may perform better for simpler or larger ANN architectures. This suggests the importance of proper coupling of different optimization algorithms; ANN architectures and methods to avoid over-fitting of real-world data – an issue that is also studied in details in the present paper. We clarified

Computer Science, Communication & Instrumentation Devices
Editors: Janahanlal Stephen, Harish Rohil and S Vasavi

ISBN: 978-981-09-5247-1

machine learning in section II, and we described the over-fitting and under-fitting as well as comparison between them in section III. At section IV and V we described the methods for avoiding over-fitting and under-fitting.

MACHINE LEARNING

The ANN learning terminates when error increases for validation data, although it often continues to decrease for training data set. When error calculated for validation data increases as that for training data decreases, it is considered as fitting to the noise present in the data instead of signal, which in other words is considered over–fitting (Sarle, W.S. 1995); (Panchal G, Ganatra A, Shah P, Panchal D. 2011). Machine-learning research has been making great progress in many directions, but this paper focuses on four (4) directions. The first direction is the improvement of classification accuracy by learning ensembles of classifiers, the second refers to the methods for scaling up supervised learning algorithms, and the third is about reinforcement learning. The fourth direction talks about the learning of complex stochastic models (Dietterich, T. G. 1997). This paper further discusses some current open problems.

OVER-FITTING AND UNDER-FITTING IN SUPERVISED LEARNING

The over-fitting

Is the one of biggest problem in training neural networks is the over-fitting of training data. That means that the neural network at the certain time during the training period does not improve its ability to solve problem anymore. But just starts to learn some random regularity contained in the set of training patterns. This is equivalent to the empirical observation that error on the test set has a minimum where the generalization ability of the network is the network is the best before this error starts to increase again (Wang, J.H.; Jiang, J.H.; Yu, R.Q. 1996); (Gaurang P, Amit G, Parth S, Devyani P. 2011); (Tom Dietterich. 1995); (Sarle, W.S. 1995); (Lawrence, S., Giles, C.L., Tsoi, A.-C. 1997). Over-fitting occurs when astatically model describes random error or nose instead of the underlying relationship (Piotrowski, A.P., Napiorkowski, J.J., 2013); (Chan, K.Y., Kwong, C.K., Dillon, T.S., Tsim, Y.C. 2011); (Panchal G, Ganatra A, Shah P, Panchal D., 2011); (Domingos, P. 2000). See Figure 1.

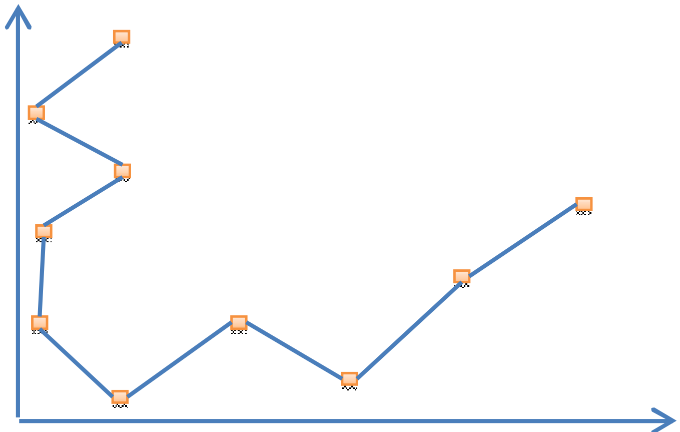

Figure 1. Over-fitting

When the model under-fits, the bias is generally high and the variance is low. Over-fitting is typically characterized by high variance, low bias estimators. In many cases, small increases in bias result in large decreases in variance (Tetko, I. V., Livingstone, D. J., and Luik, A. I. 1995); (Schaffer, C. 1993); (Loughrey, J., & Cunningham, P. 2005). See figure 2 and Figure 3.

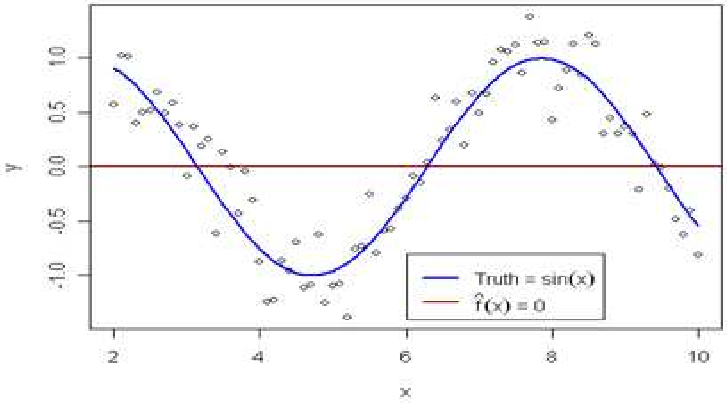

Figure 2 Zero Variance, High Bias

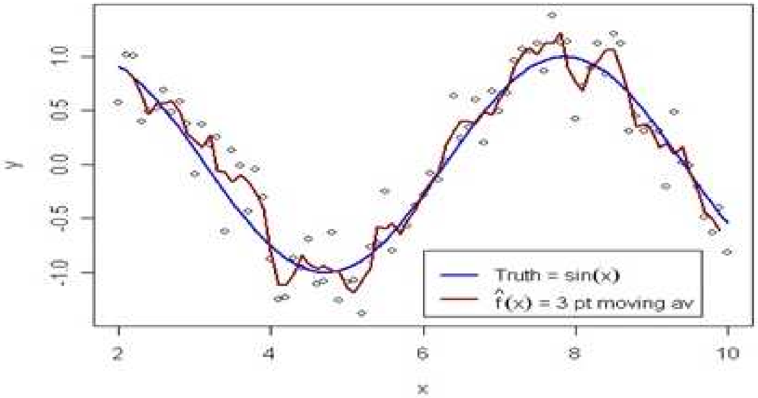

Figure 3 High Variance, Low Bias

Under-fitting

Is the opposite of Over-fitting. This occurs when the model is incapable of capturing the variability of the data. For example suppose one is training a linear (y= ax+b not polynomial a and b are constant) classifier on a data set that is a parabola Tom Dietterich. (1995), (van der Aalst, W. M., Rubin, V., Verbeek, H. M. W., van Dongen, B. F., Kindler, E., & Günther, C. W. 2010). The resultant classifier will have no predicative power nor will it able to properly map the training data (Lawrence, S., Giles, C.L., Tsoi, A.-C. 1997). This is the result of understanding or attempting to use a model which is too simple to describe a given set of data see figure 4.

Some methods to avoid the problem of over-fitting and under-fitting in supervised machine learning:

There are many methods (Kazushi. M. 2005); (Prechelt L. 1999); (Schittenkopf C, Deco G, Brauer W. 1997). :

A. Penalty methods:
   - Map provides a penalty based on P(H).
   - Minimum description length (MDL) principle.
   - Structural risk minimization.
   - Generalization cross-validation.
   - Hold and cross-validation.

B. Early stopping for training

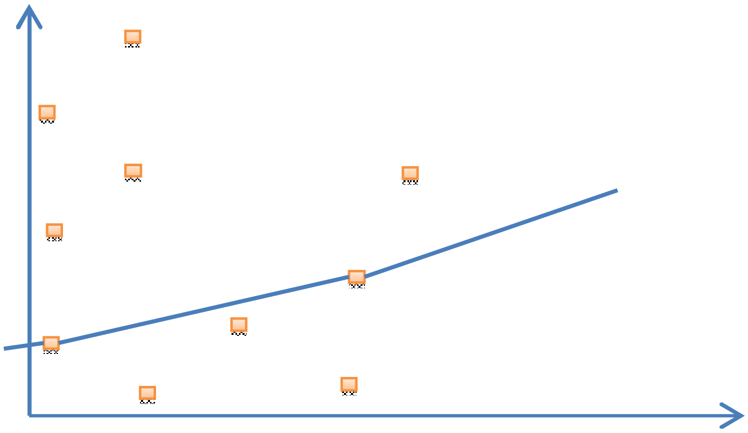

Figure 4 Under-fitting

PENALTY METHODS

Penalty method is the one of important methods to avoid the over-fitting in supervised training data, the concept of penalty it can be understand as fallow (Ruppert, D. and Carroll, R. J. 2000); (Babuška, I. 1973). :

Let $\mathbf{E}_{\text{train}}$ be our training set error and $\mathbf{E}_{\text{test}}$ be our test error.

Our real goal is to find the **h** that minimizes $\mathbf{E}_{\text{test}}$. The problem is that we can't directly evaluate $\mathbf{E}_{\text{test}}$, we can measure $\mathbf{E}_{\text{train}}$ but it is optimistic. Penalty method to find some penalty such that:

$$\mathbf{E}_{\text{test}} = \mathbf{E}_{\text{train}} + \text{Penalty} \qquad (1)$$

Where we directly penalize model complexity. We can also represent the error function as (Wen, U. P., Lan, K. M., & Shih, H. S. 2009):

$$\mathbf{E}_{\text{test}} = \mathbf{E}_{\text{train}} + \lambda \quad \text{(model complexity)} \qquad (2)$$

λ is crucial and controls the Bias-Variance Trade-off.
Theoretically the ANN as a highly non-linear model can be used for fitting of any non-linear function (web.engr.oregonstate.edu 2005). Methods using a penalty term are characterized by adding a cost term **Cλ (W)** to the error function where h denotes a learning rate and w the vector of all weights wi. Training the network by back propagation changes the weights according to the negative gradient of the error function, and therefore the derivatives are the point of interest. Usually these derivatives drive's the weights of the network to zero weight decay terms) and thereby reduce the number of free parameters they propose the cost function (Haibin, Li., Duan, Z. 2009). The advantage of penalty-term methods is that training and pruning are done in parallel. Choosing the learning rate A, however, may be tricky. We can try to avoid over-fitting by maximizing the log likelihood plus a penalty function. For an MLP with one hidden layer, a suitable penalty to add to minus the log Likelihood might be

$$\lambda_1 \sum_{k=1}^{D} \sum_{j=1}^{M} \left[w_{kj}^{(1)}\right]^2 + \lambda_2 \sum_{j=1}^{M} \left[w_j^{(2)}\right]^2 \qquad (3)$$

We need to select two constants controlling the penalty, λ1 and λ2. Setting λ1 = λ2 isn't always reasonable, since a suitable value for λ1 depends on the measurement units used for the inputs, whereas a suitable value for λ2 depends on the measurement units for the response. We might try S-fold cross-validation, but it may not work well, if each training run goes to a different local maximum. So we might use a single split into estimation and validation sets, with no re-training on the whole training set.

EARLY STOPPING

It is one method used for avoid over-fitting and under-fitting and to use early stopping approach, apart from the training data set, the testing set and the validation set that is required to define stopping criteria of the method (Prechelt L. 1999). So in this way the data divided to three parts:

1. Part for training
2. Part for validation
3. Part for testing

See figure 5.

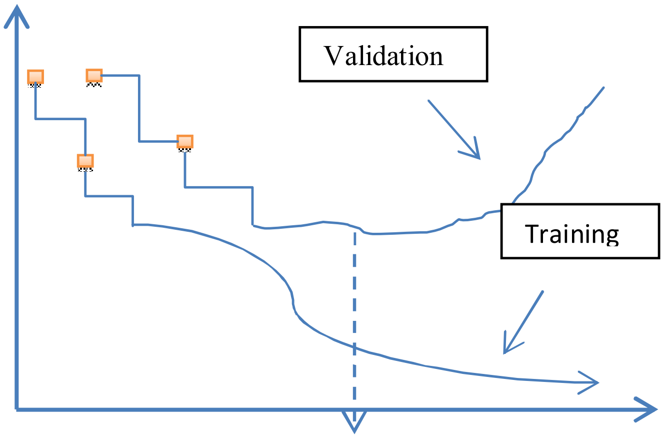

Figure. 5 The time for early stopping to avoid the over-fitting training

It is difficult to decide when it is best to stop training by just looking at the learning curve for training by itself. It is possible to over-fitting the training data if the training session is not stopped at the right point (Schittenkopf C, Deco G, Brauer W. 1997). The onset of over-fitting can be detected through cross validation in which the available data are divided into training, validation, and testing subsets. The training subset is used for computing the gradient and updating the network weights. The error on the validation set is monitored during the training session (Caruana, R., Lawrence, S., & Giles, L. (2001). The validation error will normally decrease during the initial phase of training (see Fig. 5), as does the error on the training set. However, when the network begins to over-fitting the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights at the minimum of the validation error are returned (GENCAY, R. and QI, M. 2001); (Loughrey, J., & Cunningham, P. 2005). If we have lots of data and a big model, it's very expensive to keep retraining it with different amounts of weight decay. It is much cheaper to start with very small weights and let them grow until the performance on the validation set starts getting worse (but don't get fooled by noise!) (Prechelt L. 1999). (Schittenkopf C, Deco G, Brauer W. (1997). The capacity of the model is limited because the weights have not had time to grow big. So this wills guide us to ask the question: Why early stopping works?

A. When the weights are very small, every hidden unit is in its linear range.
   - So a net with a large layer of hidden units is linear.
   - It has no more capacity than a linear net in which the inputs are directly connected to the outputs.

B. As the weights grow, the hidden units start using their non-linear ranges so the capacity grows.

It is natural to consider early stopping when dealing with big data-sets, since in this context algorithms should from optimum have computational requirements tailored to the generalization properties allowed by the data. This is exactly the main property of early stopping regularization. Figure 6 gives an illustration of this fact in a numerical simulation. Here, we added an explicit regularization parameter λ as in (Schittenkopf C, Deco G, Brauer W. 1997).

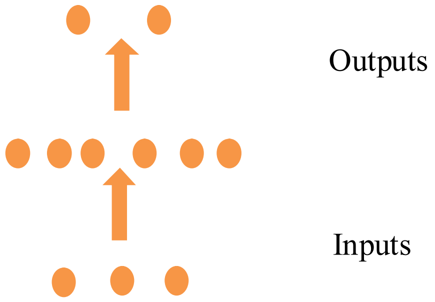

Figure 6 a numerical simulation

Early Stopping Using A Validation Set

We can tell when to stop gradient descent optimization using a set of validation cases that's separate from the cases used to compute the gradient (Prechelt, L. 1998).

Here are the steps:

- Randomly divide the training set into an estimation set and a validation set –eg, 80% of cases in the estimation set and 20% n the validation set.
- Randomly initialize the parameters to values near zero.
- Repeatedly do the following:
  i. Compute the gradient of the log likelihood using the estimation set.
  ii. Update the parameters by adding η times the gradient.
  iii. Compute the log probability of y given x for the validation cases.

  Stop when the average log probability for validation cases is substantially less than the maximum of the values found previously (definitely getting worse).
- Make predictions for test cases using the parameter values from the loop above that gave the highest average log probability to the validation cases.

Advantages of Early Stopping

Early stopping using a validation set has some advantages over a penalty method, in which you set λ1 and λ2 using S-fold cross validation, then train again on all the data with the λ1 and λ2 you choose.

With early stopping:

- You only have to train the network once, not once for each setting of λ1 and λ2 you consider, and then again on all the training data to get the final parameters.
- The performance measure from cross-validation applies directly to the actual set of network parameters you will use, not to values of λ1 and λ2 that may not do the same thing for a different local maximum.

Disadvantages of Early Stopping

Early stopping also has some problems:

- It's very ad hoc.
- It depends on details of the optimization method –eg, results could be different with a different η.
- In particular, we might want to use a different η for w(1)s than for w(2) –sort of like using different λs for a penalty method.
- Some of the training data is used only to decide when to stop this seems wasteful.

COMPARATIVE STUDY

The avoidance of over-fitting and under-fitting help to improve the performance of machine learning and for avoiding those problems. We present two of comparatives the first depend on various parameters and the second Comparative study between Over-fitting VS Under-fitting (Tetko, I. V., Livingstone, D. J., and Luik, A. I. 1995); (Prechelt L. 1999); (Tom Dietterich. 1995); (Schittenkopf C, Deco G, Brauer W. 1997); (Schittenkopf C, Deco G, Brauer W. 1997), (Caruana, R., Lawrence, S., & Giles, L. 2001); (GENCAY, R. and QI, M. 2001); (Ruppert, D. and Carroll, R. J. 2000); (Babuška, I. 1973); (Loughrey, J., & Cunningham, P. 2005); (Prechelt, L. 1998).

Comparative study between Penalty methods and early stopping

We present in this comparison between early stopping method and penalty methods as follow in table (1). A comparison based on various parameters derived from some theoretical studies.

Table 1. Comparative by methods

Methods	Penalty methods	Early stopping
Cost	Bad	Good
Speed of training	Good	Bad
Accuracy in avoidance	Good	Bad
Difficulty	Good	Bad
Capacity	Bad	Good
Time of training	Good	Bad
Influenced by variance	Bad	Good
Influenced by bias	Bad	Good
Accuracy in working with another method	Bad	Good
Small data	Bad	Good
Big data	Good	Bad

Comparative study between Over-fitting VS Under-fitting

When discussing the quality of a model in regards to a set of data two commonly used terms are over-fitting and under-fitting. A model which is over-fitted is a model which has an excess of parameters. The added complexity may and often does help the model perform well on a set of training data but it inhibits prediction of future points. Figure 2 on over-fitting illustrates this. While it is clear from the picture that we are looking at data generated by a linear function plus noise (Is the human brain not a powerful machine that it can deter- mine that on the y) the over-fitted example gains improved accuracy on the training set (the points which it learns on or the points which are drawn on the graph, while missing the overall message or pattern in the data). Noise, hidden factors, and difficult high level relations are primary causes of variability in data that cannot or is difficult to capture with statistical models. Under-fitting is the opposite of over-fitting. This occurs when the model is incapable of capturing the variability of the data. For example suppose one is training a LINEAR (y=ax+b not polynomial a and b are constants) classifier on a data set that is a parabola. The resultant classifier will have no predictive power nor will it be able to properly map the training data. This is the result of under-fitting, or attempting to use a model which is too simple to describe a given set of data. Understanding these two phenomena allows one to thread the needle and go into the space between the two extremes. It is in this gap where the model has predictive power in the validation set lies.

CONCLUSION

In this paper we performed a comparative study on various methods of machine learning to avoid over-fitting and under-fitting and we have found the early stopping method is the best as compared with penalty method that can avoid over-fitting and under-fitting with taking care to the validation time. As well as the penalty method it is sensitive to variance and bias while early stopping method it is less sensitive to variance and bias. Besides we find that under-fitting neural networks perform poorly on both training and test sets while over-fitting networks may do very well on training sets but terribly on test sets.

REFERENCES

Wang, J.H.; Jiang, J.H.; Yu, R.Q.(1996). Robust back propagation algorithm as a chemometric tool to prevent the overfitting to outliers. Chemom. Intell. Lab. Syst. 34, 109-115.

Santos, E. M., Sabourin, R., & Maupin, P. (2009). Overfitting-cautious selection of ensembles of classifiers with genetic algorithms based on complexity, diversity and accuracy criteria. Information Fusion, 10, 150–162.

Gaurang P, Amit G, Parth S, Devyani P (2011). "Determination Of Over-Learning And Over-Fitting Problem In Back Propagation Neural Network" International Journal on Soft Computing (IJSC), Vol.2, No.2.

Piotrowski, A.P., Napiorkowski, J.J., 2013. A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modeling. J. Hydrol. 476, 97–111.

Kazushi. M ,(2005). Avoiding overfitting in multilayer perceptrons with feeling-of-knowing using self- rganizing maps; 80(1):37-40.

Chan, K.Y., Kwong, C.K., Dillon, T.S., Tsim, Y.C., 2011. Reducing overfitting in manufacturing process modeling using a backward elimination based genetic programming. Applied Soft Computing 11, 1648–1656.

Tetko, I. V., Livingstone, D. J., and Luik, A. I. (1995). Neural network studies. 1. Comparison of overfitting and overtraining. J. Chem. Info. Comp. Sci., 35:826–833.

Prechelt L. (1999). Early stopping — but when? In: Orr GB, Muller OR, editors. Neural networks: Tricks of the trade. Berlin (Germany): Springer-Verlag Telos; p. 57–69.

Tom Dietterich.(1995). Overfitting and Undercomputing in Machine Learning volume {27}, pages{326--327}.

Schittenkopf C, Deco G, Brauer W. (1997). Two strategies to avoid over"tting in feedforward networks. Neural Networks;10:505}16.

Kivinen, J., Smola, A. J., & Williamson, R. C. (2004). Online Learning with Kernels. IEEE Transactions on Signal Processing, 52, 2165-2176.

Raskutti, G., Wainwright, M. J., & Yu, B. (2014). Early stopping and non-parametric regression: an optimal data-dependent stopping rule. The Journal of Machine Learning Research, 15(1), 335-366.

Dietterich, T. G. (1997). Machine-learning research. AI magazine, 18(4), 97.

Sarle, W.S. (1995). Stopped training and other remedies for over-fitting. In Proceedings of the twenty-seventh symposium on the interface of computing science and statistics (pp. 352–360).

Panchal G, Ganatra A, Shah P, Panchal D., (2011). Determination of over-learning and over-fitting problem in back propagation neural network, Int J Soft Comput. 2: 40-51.

Lawrence, S., Giles, C.L., Tsoi, A.-C. (1997). Lessons in neural network training: Overfitting may be harder than expected. In Proceedings of the fourteenth national conference on artificial intelligence, AAAl-97, (pp. 540–545). Mento Park, CA: AAAl Press.

Domingos, P. (2000). Bayesian averaging of classifiers and the overfitting problem. Proceedings of the International Conference on Machine Learning (ICML) (pp. 223–230).

Caruana, R., Lawrence, S., & Giles, L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in neural information processing systems, 402-408.

GENCAY, R. and QI, M. (2001). Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping and bagging. IEEE Trans. Neural Networks 12 726–734.

Ruppert, D. and Carroll, R. J. (2000). Spatially-adaptive penalties for spline fitting. Aust. New Z. J. Statist., 42, 205–224.

Schaffer, C. (1993). Overfitting avoidance as bias. Machine learning, 10(2), 153-178.

Babuška, I. (1973). The finite element method with penalty. Mathematics of computation, 27(122), 221-228.

Loughrey, J., & Cunningham, P. (2005). Using early-stopping to avoid overfitting in wrapper-based feature selection employing stochastic search. Trinity College Dublin, Department of Computer Science.

van der Aalst, W. M., Rubin, V., Verbeek, H. M. W., van Dongen, B. F., Kindler, E., & Günther, C. W. (2010). Process mining: a two-step approach to balance between underfitting and overfitting. Software & Systems Modeling,9(1), 87-111.

Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. Neural Networks, 11(4), 761-767.

Wen, U. P., Lan, K. M., & Shih, H. S. (2009). A review of Hopfield neural networks for solving mathematical programming problems. European Journal of Operational Research, 198(3), 675-687.

web.engr.oregonstate.edu (2005) /~tgd/classes/534/slides/part10.pdf.

Haibin, Li., Duan, Z. (2009). An L1 exact penalty function neural network method for constraint nonlinear programming problems [J]. Acta Electronica Sinica, 2009, 37(1): 229-234 (in Chinese).

BIOGRAPHY AUTHORS

Haider Khalaf Jabbar Allamy: Haider allamy, is a research scholar in the Department of Computer Science; Aligarh Muslim University, Aligarh, India. He joined to his Ph.D course in 28-09-2012. His research interest includes Artificial Intelligence and Expert system. He did B.Sc Degree in computer science from University of Basrah, College of Science, Iraq, M.Sc in computer science from Jamia Hamdard University, Delhi, India. He is envoy by the University of Misan/Iraq for complete his Ph.D course in Aligarh Muslim University, Aligarh, India.

Dr. Rafiqul Zaman Khan: Dr. Rafiqul Zaman Khan, is presently working as a Associate Professor in the Department of Computer Science at Aligarh Muslim University, Aligarh, India. He received his B.Sc Degree from M.J.P Rohilkhand University, Bareilly, M.Sc and M.C.A from A.M.U. and Ph.D (Computer Science) from Jamia Hamdard University. He has 20 years of Teaching Experience of various reputed International and National Universities viz King Fahad University of Petroleum & Minerals (KFUPM), K.S.A, Ittihad University, U.A.E, Pune University, Jamia Hamdard University and AMU, Aligarh. He worked as a Head of the Department of Computer Science at Poona College, University of Pune. He also worked as a Chairman of the Department of Computer Science, AMU, Aligarh. His Research Interest includes Parallel & Distributed Computing, Gesture Recognition, Expert Systems and Artificial Intelligence. Presently 04 students are doing PhD under his supervision. He has published about 50 research papers in International Journals/Conferences. Names of some Journals of repute in which recently his articles have been published are International Journal of Computer Applications (ISSN: 0975-8887), U.S.A, Journal of Computer and Information Science (ISSN: 1913-8989), Canada, International Journal of Human Computer Interaction (ISSN: 2180-1347), Malaysia, and Malaysian Journal of Computer Science(ISSN: 0127-9084), Malaysia. He is the Member of Advisory Board of International Journal of Emerging Technology and Advanced Engineering (IJETAE), Editorial Board of International Journal of Advances in Engineering & Technology (IJAET), International Journal of Computer Science Engineering and Technology (IJCSET), International Journal in Foundations of Computer Science & technology (IJFCST) and Journal of Information Technology, and Organizations (JITO).