Nuclear Norm Regularized Randomized Neural Network

Anupriya Gogna and Angshul Majumdar

Indraprasatha Institute of Information Technology
anupriyag@iiitd.ac.in and angshul@iiitd.ac.in

Abstract. Extreme Learning Machine (ELM) or Randomized Neural Network (RNN) is a feedforward neural network where the network weights between the input and the hidden layer are not learned; they are assigned from some probability distribution. The weights between the hidden layer and the output targets are learnt. Neural networks are believed to mimic the human brain; it is well known that the brain is a redundant network. In this work we propose to explicitly model the redundancy of the human brain. We model redundancy as linear dependency of link weights; this leads to a low-rank model of the output (hidden layer to target) network. This is solved by imposing a nuclear norm penalty. The proposed technique is compared with the basic ELM and the Sparse ELM. Results on benchmark datasets, show that our method outperforms both of them.

Keywords: Feedforward neural network, Extreme learning machine, low-rank, nuclear norm

1 Introduction

Neural networks are believed to mimic the human brain. The conventional architecture for a neural network is an input layer (for the samples), followed by a hidden layer and at the output is the target or class labels. Traditional neural network learns the link weights between the input and hidden layer nodes as well as the weights between the hidden layer nodes and the target. Randomized neural networks (RNN) or extreme learning machines (ELM) do not learn the weights between the input and the hidden layer; these weights are assigned (fixed) following some random probability distribution.

There are some studies in cognitive sciences supporting the usage of random filters in early vision; also there is mounting mathematical evidence from random matrix theory that points to linear separability [1, 2]. Basically, random projections play the same role as a non-linear kernel, it projects the data to a space such that it is linearly separable. ELM / RNN [3] is based on the same principle. However, using kernels for ELM [4, 5] seems to be an overkill, since the purpose of using a deterministic kernel and random projection is the same – linear separability; thus using the kernel on top of random projection is not likely to improve accuracy significantly.

The usual model of neural network is not sparse, there all the link weights are non-zeroes. This increases model complexity and reduces speed. The seminal work that

adfa, p. 1, 2011.

introduced sparsity into neural network learning is Lecun's Optimal Brain Damage (OBD) [6]. In this work, the link weights were iteratively pruned by thresholding the saliency of the network. Optimization has evolved significantly since the publication of OBD almost 3 decades back; currently sparsity is introduced by imposing an l_1-norm or l_0-norm on the link weights [7-9]. Sparsity has also been introduced in the ELM framework [10]; however the formulation is slightly different, it introduces sparsity in a manner similar to sparse support vector machines.

The requirement of sparsity arises from the redundancy of the network. Sparsity kills the redundant connections and keeps only the most relevant ones. However, this is not the way human brains operate. There is a large redundancy in our brain, that is why even though thousands of neurons die in our brain regularly after a certain age, we are able to carry forth all our memory and cognitive abilities without any impairment. Even in extreme situations like shock or trauma or ischemic attacks, our brain is able to recover most of its cognitive functions. All this points to the redundancy of our brain. Since modelling the human brain is the holy grail of machine learning, instead of killing the links, we propose to explicitly model the redundancy into the neural network. In principle, we believe, our proposed model will better mimic the human brain compared to existing ones.

2 Proposed Formulation

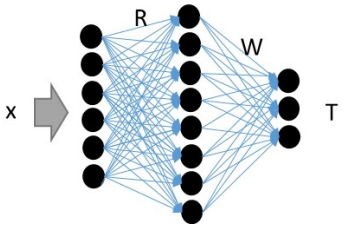

Fig. 1. Neural Network

The basic architecture for a neural network is shown above. X is the input training data. In ELMs the network weights (R) is not learnt – it is fixed. Therefore the input to the hidden layer is simply RX. There is an activation function (φ) at the hidden nodes. Therefore the output from the hidden nodes is given by,

$$Z = \varphi(RX) \tag{1}$$

The output network connects the Z to the targets T. This is given by,

$$T = WZ \tag{2}$$

Assuming a Euclidean cost function, (2) has a nice closed form solution in the form of a pseudo-inverse, given by,

$$W = \left(ZZ^T\right)^{-1} ZT^T \tag{3}$$

This solution (3) does not include any prior regarding the network weights W. The link weights are independent. We propose to incorporate redundancy; the redundancy is modeled in terms of linear dependency of the columns / rows of W. On other words this would lead to a matrix that is rank deficient. Mathematically this can be expressed as,

$$\arg\min_W \left\|T - WZ\right\|_F^2 \text{ such that W is low-rank} \tag{4}$$

Unfortunately (4) is an NP hard problem; the complexity of solving this problem is doubly exponential. Researchers in machine learning and signal processing have been interested in this problem in the past few years for a variety of applications – Collaborative Filtering [11], Distributed Sensor Network [12, 13], Direction of Arrival estimation [14] etc. What they do is to relax the NP hard rank minimization problem with their closest convex surrogate the nuclear norm, leading to,

$$\arg\min_W \left\|T - WZ\right\|_F^2 \text{ such that } \left\|W\right\|_{NN} \le \tau \tag{5}$$

Here the subscript denotes the nuclear norm, defined as the sum of the singular values of a matrix.

This (5) can be efficiently solved by a Split Bregman technique proposed in [15]. It introduces a proxy variable Y=Z and solve an augmented Lagrangian by incorporating a Bregman relaxation variable (B).

$$\arg\min_{W,Y} \left\|T - WZ\right\|_F^2 + \lambda \left\|Y\right\|_{NN} + \mu \left\|Y - Z - B\right\|_F^2 \tag{6}$$

The problem (6) can be solved using alternating directions method of multipliers (ADMM) leading to the following two sub-problems. The idea is to have sub-problems that can be solved using stock off-the-shelf algorithms.

$$\arg\min_W \left\|T - WZ\right\|_F^2 + \mu \left\|Y - W - B\right\|_F^2 \tag{7}$$

$$\arg\min_Y \lambda \left\|Y\right\|_{NN} + \mu \left\|Y - Z - B\right\|_F^2 \tag{8}$$

The first sub-problem (7) is a simple least squares problem that is solved using conjugate gradient. The second sub-problem can be efficiently updated using singular value shrinkage [16, 17]; shown as below

$$Y \leftarrow \arg\min_{Y} \lambda \|Y\|_{NN} + \mu \|Y - Z - B\|_F^2$$

$$USV^T = SVD(Z + B)$$

$$\Sigma = diag\left(\max(0, S - \tfrac{\lambda}{2\mu})\right)$$

$$Y = U\Sigma V^T$$

This concludes the derivation of the algorithm. There are two stopping criteria for the Split Bregman algorithm. Iterations continue till the objective function converges (to a local minima). The other stopping criterion is a limit on the maximum number of iterations. We have kept it to be 200.

Our method requires specification of a parameter λ and a hyper-parameter μ. In Split Bregman techniques, usually the hyper-parameter is fixed and the parameter is tuned. We follow the same routine here; we fix μ=1 and tune λ by the L-curve method.

3 Experimental Evaluation

3.1 Results on Benchmark Classification Datasets

Our experiments were carried out on some well known databases from the UCI Machine Learning repository [18]. Leave-one-out cross validation is used for avoiding variance due to random splits. Also, in order to avoid variations arising out of assignment of link weights for the first layer, the same i.i.d Gaussian random projection matrix (between the input and the hidden layer) is used for all the ELM classifiers.

Here, we compare with the basic ELM [3] and Sparse ELM [10] with linear and rbf kernels. In [10] an empirical analysis showed that for all kinds of ELM, the performance saturates when the number of hidden nodes are about 10 times the dimensionality of the vectors. Therefore we follow the same rule-of-thumb in our experiments.

Table 1. Classification Accuracy on Benhmark Datasets

Name	# classes	Basic ELM	Sparse ELM (linear)	Sparse ELM (rbf)	**Proposed**
Page Block	5	**96.86**	95.32	95.78	96.33
Abalone	29	24.22	26.49	27.39	**28.98**
Segmentation	7	95.87	96.31	**97.22**	**97.22**
Yeast	10	54.32	57.71	57.75	**59.00**
German Credit	2	75.88	75.40	76.16	**78.43**
Tic-Tac-Toe	2	86.72	85.31	85.31	**86.88**
Vehicle	4	72.97	73.46	74.51	**77.88**
Australian Cr	2	87.15	86.52	87.14	**89.64**

Balance Scale	3	85.52	93.33	94.33	**95.33**
Ionosphere	2	91.67	91.67	92.20	**94.12**
Liver	2	69.04	69.04	69.04	**70.21**
Ecoli	8	80.26	81.26	81.45	83.86
Glass	7	69.23	69.23	**70.19**	**70.19**
Wine	3	74.69	8**5.51**	85.45	85.45
Iris	3	92.00	96.00	96.67	**98.67**
Lymphography	4	88.64	86.32	86.32	**88.81**
Hayes Roth	3	34.85	41.01	43.94	**45.38**
Satellite	6	89.73	80.30	83.15	**86.22**
Haberman	2	65.22	63.28	63.20	**67.78**

Experimental results show that our method yields the best results, except in one dataset (Wine), where the sparse ELM with linear kernel yields the best results. One interesting observation that can be made here is that adding a non-linear kernel to the ELM formulation does not help much; one can see that the difference between the linear and rbf kernel sparse ELM is not much different (less than 1%). This phenomenon has been explained before – both random projections and non-linear kernel randomize make the data linearly separable, hence adding one to of the other does not change much. It must be noted, this observation is not available in the original paper for sparse ELM since they had not compared with linear kernels.

3.2 Experiments on Face Recognition

Fig. 2. Samples from Extended YaleB

We follow the experimental protocol outlined in [19]. The experiments are carried on the Extended Yale B Face Database. For each subject, we randomly select half of the images for training and the other half for testing. Table 2 contains the results for face recognition. The features are selected using the simple Eigenface method. Although more sophisticated feature extraction techniques exist, our goal is to investigate that given the feature set how different classifiers perform. To compare our results with [19], we select the same number of Eigenfaces as proposed there in.

We do not compare the results with SVM and ANN, since it has already shown in [19] that the SRC (sparse representation based classification) outperforms them for face recognition problems. We compare our results with basic ELM, and sparse (rbf kernel) ELMs as well. As before, in order to avoid variations due to random assign-

ment of the link weights between the input and the hidden layer, the same random projection matrix (i.i.d Gaussian) is used for all the different types of ELM classifiers.

Table 2. Face Recognition

Method	Number of Eigenfaces			
	30	56	120	504
ELM	86.49	91.71	93.87	96.77
Sparse ELM	86.96	92.05	94.26	97.13
SRC	**89.40**	**93.37**	**95.14**	**97.79**
Proposed	87.11	92.56	95.08	97.25

SRC is a lazy learning classifier; it has no training time but a large testing time since it requires solving a sophisticated optimization problem. Our proposed method cannot beat SRC but yields better results than the other ELM variants.

3.3 Experiments on Handritten Digit Recognition

The MNIST digit classification task is composed of 28x28 images of the 10 handwritten digits. There are 60,000 training images with 10,000 test images in this benchmark. The images are scaled to [0,1] and we do not perform any other pre-processing. However, we do not carry out experiments on the standard MNIST data-set; experiments are also carried out on the more challenging variations of the MNIST dataset [20]. These were introduced as benchmark deep learning datasets. All these datasets have 10,000 training, 2000 validation and 50,000 test samples. The size of the image as before is 28x28 and the number of classes are 10.

Dataset	Description
basic-rot	Smaller subset of MNIST with random rotations.
bg-rand	Smaller subset of MNIST with uniformly distributed random noise background.
bg-img	Smaller subset of MNIST with random image background.
bg-img-rot	Smaller subset of MNIST digits with random background image and rotation.

As before we compare our proposed technique with the basic ELM and the sparse ELMs with linear and rbf kernels. The results are shown in Table 3. We want to eliminate effects arising out of random assignment of link weights in the first layer; in order to do so, we use the same random projection matrix (i.i.d Gaussian) for all the classifiers.

From Table 3, as expected, our proposed technique yields significantly better results than the others.

Table 3. Digit Classification

Dataset	ELM	Sparse ELM (linear)	Sparse ELM (rbf)	Proposed
basic	92.79	93.16	93.89	**95.08**
basic-rot	86.70	87.47	86.70	**88.21**
bg-rand	86.06	86.70	90.27	**90.94**
bg-img	80.69	80.32	80.69	**82.59**
bg-img-rot	52.61	56.24	52.61	**54.58**

4 Conclusion

In this work we have proposed a variation for randomized neural network / extreme learning machine. Prior variants of the basic technique included kernels and sparsity. In sparsity based techniques the redundant connections are pruned; only the most relevant ones stay. In this work, our goal is to better mimic the human brain. Therefore instead of pruning the connections, we actively promote redundancy in the system. This is achieved by modelling redundancy in terms of linear dependency. In turn, this leads to a low-rank representation of the matrix containing the link weights between the hidden layer and the targets. Following signal processing literature, we formulate a nuclear norm regularized ELM problem. Efficient solutions for this problem already exist.

We carry out thorough experimental validation. We validate on 1. benchmark machine learning datasets from the UCI Machine Learning Repository; 2. Face Recognition (YaleB) and 3. Handrwitten digit recognition (MNIST variations). In all the cases, our method outperforms the basic ELM and the sparse ELM (linear and rbf).

5 References

1. S. Paul, C. Boutsidis, M. Magdon-Ismail, P. Drinea, “Random Projections for Linear Support Vector Machines”, ACM Transactions on Knowledge Discovery from Data, Vol. 8 (4), 2014
2. Q. Shi, C. Shen, R. Hill, A. van den Hengel, “Is margin preserved after random projection?”, ICML, 2012.
3. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: Theory and applications,”Neurocomputing, Vol. 70, pp. 489–501, 2006.

4. S. Scardapane, D. Comminiello, M. Scarpiniti and A. Uncini, “Online Sequential Extreme Learning Machine With Kernels”, IEEE Transactions on Neural Networks and Learning Systems, Vol. 26 (9), pp. 2214-2220, 2015.
5. Y. Zhou, J. Peng and C. L. P. Chen, “Extreme Learning Machine With Composite Kernels for Hyperspectral Image Classification”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 8 (6), pp. 2351-2360, 2015.
6. Y. LeCun, “Optimal Brain Damage”, NIPS, 1990.
7. M. Thom and G. Palm, “Sparse Activity and Sparse Connectivity in Supervised Learning”, Journal of Machine Learning Research, Vol. 14, pp. 1091-1143, 2013.
8. V. Gripon, “Sparse Neural Networks With Large Learning Diversity”, IEEE Transactions on Neural Networks and Learning Systems, Vol. 22 (7), pp. 1087-1096, 2011.
9. X. Glorot, A. Bordes and Y. Bengio, “Deep Sparse Rectifer Neural Networks”, AISTATS 2011.
10. Z. Bai, G.-B Huang, D. Wang, H. Wang, and M. B. Westover, “Sparse Extreme Learning Machine for Classification”, IEEE Transactions on Cybernetics, Vol. 44 (10), pp. 1858-1870, 2014.
11. A. Gogna and A. Majumdar, “Matrix Completion Incorporating Auxiliary Information for Recommender System Design”, Expert Systems with Applications, Vol. 42 (5), pp. 5789-5799, 2015
12. A. Majumdar and R. K. Ward, “Increasing Energy Efficiency in Sensor Networks: Blue Noise Sampling and Non-Convex Matrix Completion”, International Journal of Sensor Networks, Vol. 9, (3/4), pp. 158-169, 2011.
13. A. Jindal and K. Psounis, “Modeling Spatially Correlated Data in Sensor Networks”, ACM Transactions on Sensor Networks, Vol. 2 (4), pp. 466–499, 2006.
14. P. Pal and P. P. Vaidyanathan, “A Grid-Less Approach to Underdetermined Direction of Arrival Estimation Via Low Rank Matrix Denoising”, IEEE Signal Processing Letters, Vol. 21 (6), pp. 737-741, 2014.
15. A. Gogna, A. Shukla and A. Majumdar, “Matrix Recovery using Split Bregman”, International Conference on Pattern Recognition, 2014
16. A. Majumdar and R. K. Ward, “Some Empirical Advances in Matrix Completion”, Signal Processing, Vol. 91(5), pp. 1334-1338, 2011
17. R. Chartrand, "Nonconvex splitting for regularized low-rank + sparse decomposition", IEEE Transactions on Signal Processing, Vol. 60, pp. 5810-5819, 2012.
18. http://archive.ics.uci.edu/ml/
19. J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31 (2), pp. 210 – 227, 2009.
20. http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/MnistVariations