Combining ELMs with Random Projections

IEEE Intelligent Systems 28(6), pp. 46-48, 2013

Paolo Gastaldo and Rodolfo Zunino (University of Genoa),
Erik Cambria (MIT Media Laboratory), Sergio Decherchi (Italian Institute of Technology)

In the Extreme Learning Machine (ELM) model [1], a Single-hidden-Layer Feedforward Network (SLFN) implements inductive supervised learning by combining two distinct components. A hidden layer performs an explicit mapping of the input space to a feature space; the mapping is not subject to any optimization, since all the parameters in the hidden nodes are set randomly. The output layer includes the only degrees of freedom, i.e., the weights of the links that connect hidden neurons to output neurons. Thus, training just requires one to solve a linear system by a convex optimization problem. The literature proved that the ELM approach can attain a notable representation ability [1].

According to the ELM scheme, the configuration of the hidden nodes ultimately defines the feature mapping to be adopted. Actually, the ELM model can support a wide class of activation functions. Indeed, an extension of the ELM approach to kernel functions has been discussed in the literature [1].

This paper addresses the specific role played by feature mapping in ELM. The goal is to analyze the relationships between such feature mapping schema and the paradigm of *random projections* (RP) [2]. RP is a prominent technique for dimensionality reduction that exploits random subspaces. This research shows that RP can support the design of a novel ELM approach, which combines generalization performance with computational efficiency. The latter aspect is attained by the RP-based model, which always performs a dimensionality reduction in the feature mapping stage, and therefore shrinks the number of nodes in the hidden layer.

ELM feature mapping

Let $\mathbf{x} \in \Re^d$ denote an input vector. The function, $f(\mathbf{x})$, of an output neuron in an ELM that adopts L 'hidden' units is written as

$$f(\mathbf{x}) = \sum_{j=1}^{L} w_j \cdot a(\mathbf{r}_j \cdot \mathbf{x} + b_j) \tag{1}$$

Thus, a set of random weights $\{\mathbf{r}_j \in \Re^d;\ j=1,..,L\}$ connects the input to the hidden layer; the j-th hidden neuron embeds a random bias term, b_j, and a nonlinear *activation* function, $a(\cdot)$. A vector of weighted links, $\mathbf{w} \in \Re^L$, connects the hidden layer to the output neuron.

The vector quantity $\mathbf{w} = [w_1, .., w_L]$ embeds the degrees of freedom in the ELM learning process, which can be formalized after introducing the following notations:

- $\mathbf{X}$ is the $N \times (d+1)$ matrix that originates from the training set. $\mathbf{X}$ stems from a set of N labeled pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is the i-th input vector and $y_i \in \Re$ is the associate expected 'target' value.
- $\mathbf{R}$ is the $(d+1) \times L$ matrix with the random weights.

Here, by using a common trick, both the input vector, $\mathbf{x}$, and the random weights, $\mathbf{r}_j$, are extended to $\mathbf{x}$: $=[x_1,.., x_d, 1]$ and $\mathbf{r}_j \in \Re^{d+1}$ to include the bias term.

Accordingly, the ELM learning process requires one to solve the following linear system

$$\mathbf{y = Hw} \tag{2}$$

where $\mathbf{H}$ is the hidden layer output matrix obtained by applying the activation function, $a()$, to every element of the matrix:

$$\mathbf{XR} \tag{3}$$

Expression (3) clarifies that in the ELM scheme (1) the hidden layer performs a mapping of the original d-dimensional space into a L-dimensional space through the random matrix $\mathbf{R}$, which is set independently from the distribution of the training data. In principle, the feature mapping phase may either involve a reduction in dimensionality ($L < d$) or, conversely, remap the input space into in an expanded space ($L > d$).

Both theoretical and practical criteria have been proposed in the literature to set the parameter L [1, 3]. This quantity is crucial because it determines the generalization ability of the ELM. At the same time, it affects the eventual computational complexity of both the learning machine and the trained model. These aspects become critical in hardware implementations of the ELM model, where resource occupation is of paramount importance.

A few pruning strategies for the ELM model have been proposed in the literature to balance generalization performance and computational complexity [3]. The present work tackles this problem from a different perspective and proposes to exploit the fruitful properties of random projections.

The approach discussed here applies RP to reduce the dimensionality of data; the study, however, opens interesting vistas on using RP to tune the basic quantity, L, as well.

Dimensionality reduction by using RP

RP is a simple and powerful dimension reduction technique that uses a suitably scaled random matrix with independent, normally distributed entries to project data into low-dimensional spaces. The procedure to get a random projection is straightforward and arises from the Johnson-Lindenstrauss (JL) lemma [2]. The lemma states that any N point set lying in d-dimensional Euclidean space can be embedded into a r-dimensional space, with $r \geq O(\varepsilon^{-2}\ln(N))$, without distorting the distances between any pair of points by more than a factor $1 \pm \varepsilon$, where $\varepsilon \in (0, 1)$.

Over the years, the use of probabilistic methods greatly simplified the original proof of Johnson and Lindenstrauss, and at the same time lead to straightforward randomized algorithms for implementing the transformation. In matrix notation, the embedding operation is expressed as

$$\mathbf{K} = \mathbf{XP} \tag{4}$$

where $\mathbf{X}$ is the original set of N, d-dimensional observations, $\mathbf{K}$ is the projection of the data into a lower, r-dimensional subspace, and $\mathbf{P}$ is the random matrix providing an embedding that satisfies the JL lemma.

In principle, (4) is a projection only if $\mathbf{P}$ is orthogonal; this ensures that similar vectors in the original space remain close to each other in the low-dimensional space. In very high-dimensional spaces, however, bypassing orthogonalization saves computation time without affecting the quality of the projection matrix significantly. In this regard, the literature provides a few practical criteria to build $\mathbf{P}$ [2].

RP-ELM

The ability of RP to preserve, approximately, the distances between the N data vectors in the r-dimensional subspace is a valuable property for machine learning applications in general [4]. Indeed, this property is the conceptual basis of the novel approach that connects the ELM feature mapping scheme (3) to the RP paradigm.

A new ELM model can be derived from (1) if one set as hypotheses that 1) L should be smaller than d and 2) the mapping implemented by the weights $\mathbf{r}_j$ satisfies the JL lemma. Under these assumptions, the mapping scheme (3) always implements the dimensionality reduction process (4). In practice, one takes advantage of the properties of RP to obtain an ELM model that shrinks the size L of the hidden layer and reduces the computational overhead accordingly. The eventual model will be denoted as "RP-ELM." The crucial point is that the JL lemma guarantees that the original geometry of the data is only slightly perturbed by the dimensionality reduction process [2]; indeed, the degradation grows gradually as L decreases (given d and N) [2].

In principle, the literature provides several criteria for the construction of a random matrix that satisfies the JL lemma. The present work focuses on matrices where the entries are independent realizations of ±1 Bernoulli random variables [2]; hence, matrix $\mathbf{R}$ (3) is generated as follows:

$$\mathbf{R}_{i,j} = \begin{cases} 1/\sqrt{L} & \text{with probability } 1/2 \\ -1/\sqrt{L} & \text{with probability } 1/2 \end{cases} \tag{5}$$

Baraniuk et al. [2] has showed that this kind of random matrices actually satisfies both the JL lemma and the *restricted isometry property*, thus bringing out a connection between RP and compressed sensing.

Experimental Results

The performance of the proposed RP-ELM model has been tested on two binary classification problems [5]: *Colon Cancer* and *Leukemia*. The former dataset contains expression levels of 2,000 genes taken in 62 different samples; 40 samples refer to tumour samples. The latter dataset provides the expression levels of 7,129 genes taken over 72 samples; 25 samples refer to "acute lymphoblast leukemia" and 47 samples refer to "acute myeloid leukemia." The datasets share two interesting features: 1) the number of patterns is very low, and 2) the dimensionality of data is very high as compared with the number of patterns. In both cases, data are quite noisy, since gene expression profiles are involved.

The experimental session aimed to evaluate the ability of the RP-ELM model to suitably trade-off generalization performance and computational complexity (i.e., number of nodes in the hidden layer). It is worth noting that the experiments did not address gene selection. Table 1 reports on the results of the two experiments, and gives the error rates attained for ten different settings of L. In both cases, the highest values of L corresponded to a compression ratio of 1:20 in the feature-mapping stage. The performances were assessed by adopting a Leave-One-Out (LOO) scheme, which yielded the most reliable estimates in the presence of limited-size dataset. Error rates were worked out as the percentage of misclassified patterns over the test set.

Table 1. Error rates scored by RP-ELM and standard ELM on the two binary classification problems

Colon Cancer			Leukemia		
	Error Rate (%)			Error Rate (%)	
L	RP-ELM	ELM	*L*	RP-ELM	ELM
10	38.7	38.7	35	25.0	40.3
20	40.3	35.5	70	27.8	31.9
30	43.5	45.2	105	47.2	27.8
40	32.3	45.2	140	30.6	33.3
50	29.0	50.0	175	37.5	37.5
60	37.1	48.4	210	25.0	37.5
70	37.1	40.3	245	27.8	40.3
80	29.0	37.1	280	31.9	36.1
90	29.0	43.5	315	31.9	30.6
100	25.8	40.3	350	38.9	33.3

The table compares the results of the RP-ELM model with those attained by the standard ELM model. Results showed that, in both experiments, RP-ELM attained lower error rates than standard ELM. Moreover, the RP-ELM performed comparably with approaches reported in the literature, in which ELM models included 1,000+ neurons and did not adopt a LOO validation procedure.

Conclusions

The paper introduced a novel model for ELMs that exploits RP techniques. Theory showed that, by a direct implementation of the JL lemma, one can sharply reduce the number of neurons in the hidden node without affecting the generalization performance in prediction accuracy. As a result, the eventual learning machine always benefits from a considerable simplification in the feature-mapping stage. This allows the RP-ELM model to properly balance classification accuracy and resource occupation. The experiments showed that the proposed model can attain satisfactory performance. Further investigations will aim to confirm the effectiveness of the RP-ELM scheme by additional theoretical insights and a massive campaign of experiments.

References

[1] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme Learning Machine for Regression and Multiclass Classification,” IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, vol. 42, no. 2, pp. 513-529, 2012.

[2] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A Simple Proof of the Restricted Isometry Property for Random Matrices,” Constr. Approx., vol. 28, pp. 253–263, 2008

[3] G.-B. Huang, D. H. Wang, and Y. Lan, “Extreme Learning Machines: A Survey,” International Journal of Machine Leaning and Cybernetics, vol. 2, no. 2, pp. 107-122, 2011

[4] Y. Miche, B. Schrauwen, and A. Lendasse, “Machine Learning Techniques based on Random Projections,” Proc. of European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning, ESANN 2010, Bruges (Belgium), 28-30 April 2010, pp 295-302

[5] LIBSVM Data Repository, available at: http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html