A Regularization Network Committee Machine of isolated Regularization Networks for Distributed Privacy Preserving Data Mining

Yiannis Kokkinos and Konstantinos Margaritis*

Parallel and Distributed Processing Laboratory, Department of Applied Informatics, University of Macedonia, 156 Egnatia str., P.O. Box 1591, 54006, Thessaloniki, Greece

Keywords. Distributed processing, Regularization Networks, Committee Machines, privacy preserving, data mining.

Abstract. In this paper we consider large scale distributed committee machines where no local data exchange is possible between neural network modules. Regularization neural networks are used for both the modules as well as the combiner committee in an embedded architecture. After the committee training no module will know anything else except its own local data. This privacy preserving obligation is a challenging problem for trainable combiners but crucial in real world applications. Only classifiers in the form of binaries or agents can be sent to others to validate their local data and sent back average classification rates. From this fully distributed and privacy preserving mutual validation a coarse-grained matrix can be formed to map all members. We demonstrate that it is possible to fully exploit this mutual validation matrix to efficiently train another regularization network as a meta learner combiner for the committee.

1 Introduction

A committee machine [1] exhibits an intrinsically parallel and distributed architecture [2] in which multiple modules of independently trained neural networks are combined for the same task. An ensemble learner of this kind is an ideal candidate for data mining large scale physically distributed data repositories in institutions/organizations or Peer-to-Peer networks. However privacy preserving and scalability are crucial issues for these real life applications. The Regularization Networks [3][4][5] are kernel based classifiers known to use as hidden neurons the real training data points, to form the kernel functions and capture the data closeness approximate of the underlined problem distribution. Using real points is valuable when data features have discrete values, e.g., in cases of image processing, computer vision [1] and data mining [8], a fact that elevates such type of kernel based ridge regression methods [9] [10] to state of the art. For these distributed Regularization Network (RN) modules if another Regularization Network can be trained to act as a high level combiner then can serve as a meta-learner in a distributed data mining system [11]. Such a meta-learner is looked at in detail here for the distributed privacy preserving case.

A committee of neural networks has excellent generalization capabilities, [6] since typically the committee error is reduced considerably by taking the average error of the combined networks. All neural networks classifiers are first trained in parallel based on local data to construct local data models. Then the committee combines [7] all individual decisions through proper weights to form the global data model used for collective decisions. Committees can be used in data mining physically distributed data repositories as well as peer-to-peer systems. Gather large volumes of distributed data to a single location for centralized data mining is usually unfeasible. The causes that prevent this lay in technical issues like limited network bandwidth and enormous main memory demands, practical issues like huge required training times, algorithmic issues in where mining algorithms operate only on data in main memory, and especially privacy preserving concerns that restrict the transferring of sensitive data.

To this context the task to build a global model from data distributed over workstations, without moving or sharing local data itself, and with little centralized coordination, is challenging. The trainable combiner requires a separate test set to find proper weights for the neural network modules. Thus it requires the use of extra information from their input-output mappings. At least two-by-two the classifiers must share either input vectors, or output vector results with respect to instances of an independent test set. Without exposing data between modules or without aggregating a portion of data this is tricky. A regularization network as a meta-learner committee of regularization networks trained by a simple distributed privacy preserving mutual validation matrix is presented here, as an effort to the solution of the problem.

2 Distributed Privacy-Preserving Data Mining

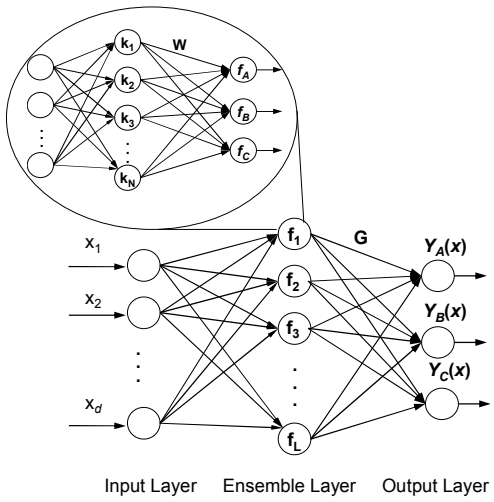
Distributed privacy-preserving data mining is the study of how to extract globally interesting models, associations, classifiers, clusters, and other useful aggregate statistics from distributed data without disclosing private information within the different participants. Data exchange and free flow of information is frequently prohibited by legal obligations or by commercial and personal concerns, since the participants may wish to collaborate, but might not fully trust each other. The basic idea of a secure multiparty computation is that a distributed computation is secure if at the end no party knows anything except its own input and the aggregate results. For example, secure sum protocol [12] computes the sum of a collection of numbers without revealing anything but the output sum. Classifiers which need total sums like Naive Bayes can be worked in this fashion. Data sets are usually distributed in horizontal partitions where different sites contain different sets of records with the same attributes. Classifier examples that have been generalized to this distributed privacy preserving data mining problem are the Naïve Bayes Classifier [13][14], the SVM Classifier with nonlinear kernels [15] and the k-nearest neighbour classifier [16]. Since for multiclass problems classical neural networks are proven the best over the years, here we present an embedded architecture Regularization Network committee machine of Regularization Networks distributed over workstations, of which the training leave the processor nodes with no extra knowledge for the other participant inputs.

3 A Regularization Networks Committee Machine

The committee training [17][18][19], like in neural network training has to find a proper weight for each individual neural network. A Regularization Network (RN) committee of embedded Regularization Networks classifiers illustrated in fig.1, is analysed in this section. An individual RN [3] [4] [5] has one input layer, one hidden layer, and one output layer. All real data points are loaded to the hidden neurons to form the kernel functions. For a training set $\{\vec{x}_i, y_i\}_{i=1}^N$, a kernel function $k(\cdot,\cdot)$, usually a Gaussian, and a Kernel matrix K with $K_{i,j} = k(\vec{x}_i, \vec{x}_j)$, the RN training phase finds optimum weights *w* for the output f() by solving in Reproducing Kernel Hilbert Space H_K a minimization problem for a regularized functional which consists of a usual data term plus a second regularization term that plays the role of the stabilizer [3] [4] [5]

$$\arg\min_{f \in H_K} \left\{ \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\vec{x}_i))^2 + \gamma \| f \|_K^2 \right\} \tag{1}$$

For a class C the weights w_C is the solution of a linear system $(K + N\gamma I)w_C = y_C$, where I is identity matrix, K is the kernel matrix, $\gamma > 0$ is the regularization parameter and $y_C = (y_1, \ldots, y_N)$ are the desired output labels, 1 for class C and 0 for the others.



$$f_{i,A}(\vec{x}) = \sum_{n=1}^{N_i} w_{A,n} \cdot k(\vec{x}_n, \vec{x})$$

$$f_{i,B}(\vec{x}) = \sum_{n=1}^{N_i} w_{B,n} \cdot k(\vec{x}_n, \vec{x})$$

$$f_{i,C}(\vec{x}) = \sum_{n=1}^{N_i} w_{C,n} \cdot k(\vec{x}_n, \vec{x})$$

$$Y_A(\vec{x}) = \sum_{i=1}^{L} g_{A,i} \cdot f_{i,A}(\vec{x})$$

$$Y_B(\vec{x}) = \sum_{i=1}^{L} g_{B,i} \cdot f_{i,B}(\vec{x})$$

$$Y_C(\vec{x}) = \sum_{i=1}^{L} g_{C,i} \cdot f_{i,C}(\vec{x})$$

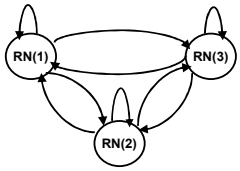
$$class(\vec{x}) = \arg\max_j \sum_{i=1}^{L} g_{j,i} \cdot f_{i,class=j}(\vec{x})$$

Fig. 1. Architecture of a Regularization Network committee machine of embedded regularization networks for the three class problem.

For the three class problem in fig.1 a Regularization Network module consists of a hidden layer with *N* neurons of kernel units and three linear outputs. $f_{i,class=j}(\vec{x})$ is the output of the RN module i for class j. The RN committee machine of L embedded RNs has also three outputs. While the training of local RNs can provide the weight vectors W_A, W_B and W_C, one for each class output, the high level weight vectors G_A, G_B and G_C for the outer regularization network committee are still unknown.

4 The Distributed mutual validation matrix

To find the weight vectors G_A, G_B and G_C in fig.1 without reveal the training data vectors between modules we must first introduce in this section the mutual validation matrix S. Assume an ensemble of three hidden Regularization Networks namely RN(1), RN(2) and RN(3) that are trained independently from each other based on their local datasets. For privacy reasons the different locations cannot contribute or share any even smallest part of their data to other modules. Only RN networks in the form of binaries or agents can be sent to other modules to validate their data. Then schematically one can work with measures of classification rates between RN(1), RN(2) and RN(3) train patterns. The RN(1) classifies its own train data points to produce 1-1 measure. Likewise RN(2) classifies its own data to produce 2-2 measure and RN(3) produce 3-3 measure. These three are internal (or intra) measures, as they controlled by internal characteristics. In consequence RN(1) classifies train data of RN(2) to produce 1-2 measure and RN(2) classifies RN(1) data to produce 2-1 measure. In the same manner the measures 1-3, 3-1, 2-3 and 3-2 are produced. These six asymmetric measures are local (or inter) as they are based on the performance of neighbours data. A coarse-grained mutual validation matrix can be filled with these average rates. The validation set for one classifier is the train set of the other and vice versa. This mutual validation matrix maps the RN members is illustrated in fig. 2.



$$S = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}$$

Fig. 2. (A) An ensemble of three different Regularization Networks RN(1), RN(2) and RN(3), inter-connected with each other via accuracy measures, (B) the mutual validation matrix.

The paradigm in fig. 2 is an illustrative example of the simple point-to-point communications involved. The diagonal measures of the matrix are the self-validation average positive hits of each RN. Upon receiving a Regularization Network i the module j apply it to classify its own local data and send back a simple average learning rate equal to $S(i, j)$ = PositiveLocalHits(j)/LocalTrainSize(j), where positive hits are the number of correctly classified local samples of module j and local train size is their number N_j of training points. In this way privacy preserving is achieved.

Distributed computations are required for the i-j, and j-i asymmetric measures between different RNs across the communication network. An asynchronous cycle is continually executed composed of commands, like sent local classifier, check for received classifier, compute local positive hits, and sent average. In terms of parallel and distributed computing this RN committee training approach is hybrid task as well as data parallel. Tasks are the RN classifiers which travel across the communication network and when arrive in a processing node are applied to the node's local data.

In this way one manages for the RN committee training to be transformed to a fully asynchronous embarrassingly parallel programming paradigm like the iterative decomposition [20]. Iterative decomposition occurs when a loop parallel execution can be done in some independent and unconnected manner. The work is statically decomposed, but the work assignments are dynamically distributed to processes. Each node may operate independently and communicate its own results to another node, making it an appropriate choice for various types of asynchronous cycles.

5 Training the Committee via the mutual validation matrix

From the distributed and data privacy preserving mutual validation the coarse-grained validation matrix formed can map all modules. Although so far such a matrix was ignored, we demonstrate here that it is possible to fully exploit it to efficiently train another regularization network as a meta learner combiner for the committee.

Recall now that the weights per class output of each local Regularization network module i of the ensemble are given by solving a linear system of the form $(K + \lambda I)w = y$, where $\lambda=N_i\gamma$. While the weight vectors G_A, G_B and G_C for the outer regularization network committee machine are unknown, the simple mutual validation matrix S can now enter into the training procedure. The weights G_A, G_B and G_C can be found by considering the matrix S as the outer kernel and solving the linear equations, one for each class, in terms of the vectors Y_A, Y_B and Y_C.

$$\begin{aligned} G_A &= (S + \lambda I)^{-1} Y_A \\ G_B &= (S + \lambda I)^{-1} Y_B \\ G_C &= (S + \lambda I)^{-1} Y_C \end{aligned} \tag{2}$$

The vectors Y_A, Y_B and Y_C correspond to classes A, B and C have all size equal to the number L of ensemble modules. A value $Y_A(i)$ is positive local hits of RN classifier i per overall train size, produced by applying the i^{th} RN module to the A class portion of its own i^{th} dataset (if not any set 0). Respectively $Y_B(i)$ is the overall classification rate produced by the i^{th} RN module for the B class portion (if not any set 0) of its data, and $Y_C(i)$ similarly for the C class portion. The regularization parameter is again λ, and I is again the identity matrix.

Besides the simplicity of this method another interesting observation is that if the local neural network classifiers all are reduced to have only a single train example then, the mutual validation matrix reduces to a usual kernel matrix and the proposed committee machine in fig.1 switches to the single conventional Regularization Network. For systems where data movement across local sites is hindered like those studied at this point the above observation can serve only as a proof of correctness. However simply means that the more fine-grained the modules are, the more accurate the committee could become. Fine-grained modules can be accomplished by globally finding compact dense clusters of data points and training all the RN modules based on these data clusters. Then this RN committee of RNs method can be possibly extended for large kernel ridge regression approximation in open systems.

6 Experimental Results

The set of experiments aims at discovering the classification performance of the RN distributed privacy preserving committee tested on a separate test set of points. To this end several benchmarks are used, taken from the UCI machine learning data repository. We compare our method against majority voting. To show the efficiency of the method we must create highly unevenly and without stratification data partitions, otherwise accurate estimations may emerge simply from the fact that we use an ensemble. For the same reason a comparison with the majority voting rule is also done.

The experimental design is as follows:

1. A dataset is randomly split into a train set (70%) and a test set (30%) with stratification.
2. The train set is distributed unevenly, randomly and without stratification across a number of processors.
3. Every processor trains a local Regularization Neural Network classifier.
4. An asynchronous computing cycle is executed to find all entries of the proposed mutual validation matrix.
5. The high level RN committee is trained using the mutual validation matrix.
6. The final RN committee is tested on the test set.
7. This procedure is repeated 10 times for each benchmark dataset and each corresponding processor number, and the error rate results are averaged.
8. A single Regularization Neural Network is trained again on the same initial train set and tested on the same test set for comparison

In step 2 the uneven as well as random and without stratification choice of a particular processor's data is important for the experiment to simulate a real situation and to show the power of the method. To this end we allow a quarter of processors to randomly peek a population size between 5 and 300 train points. Likewise another quarter randomly peek a population size between 5 and 100. Similarly the remainder half of processors are allowed to have a size between 5 and 30. Then according to the total number of training points these population sizes are normalized, in favour of the smaller ones, for their sum to fit the total. This method produces a fairly uneven unstratified distribution, with half of processors populations being small. Many of them end up with no samples from some class. In addition small local populations are likely to produce singularities to the mutual validation matrix inversion, in order to make harder the proposed training method and show the benefits of the RN stabilizer. Other uneven and irregular distributions we try have worked as well as the former one.

On all tested datasets the RN committee outperforms majority voting. The Iris dataset has 150 examples, 4 input features and 3 classes. The Diabetes dataset has 768 examples, 8 input features and 2 classes. The Wisconsin breast cancer dataset has 683 examples, 9 input features and 2 classes. The Vehicle dataset has 846 examples, 18 features and 4 classes. The Glass dataset has 214 examples, 9 input features and 6 classes.The Wine dataset has 178 examples, 13 input features and 3 classes. Although the uneven un-stratified splitting produces highly irregular data distributions, the RN committee was found to perform better not only than majority voting but also slightly better than the single RN on datasets like the Iris, Wine and Wisconsin.

Table 1. Iris dataset results

RN modules	Majority Voting error	RN committee error	single RN error
10	5,2%	3,1%	3,3%
11	8,3%	2,7%	3,1%
12	6,3%	2,7%	2,7%
13	5,8%	2,5%	2,7%
14	4,2%	3,8%	2,5%
15	4,2%	3,5%	2,9%

In table 1, RN modules column indicates the number of RNs in the ensemble layer, and is used to show the range of the applicability. All error rates are measured by the ratio (falsely classified samples)/(total) on the test set. In the second column all module networks in the ensemble layer perform simple majority voting to produce error rate. The third column shows the proposed distributed privacy preserving RN committee error. The fourth column shows the single RN error when trained on the whole train set. The RN committee outperforms the majority voting and unexpectedly recovers the single RN error rate in most of the experimental cases. A marginally better performance of RN committee over the single RN is also present in some cases.

Table 2. Diabetes dataset results

RN modules	Majority Voting error	RN committee error	single RN error
50	30,2%	26,7%	25,9%
55	29,3%	25,4%	25,4%
60	29,7%	26,7%	24,6%
65	31,5%	25,9%	25,0%
70	32,3%	25,0%	24,1%
75	31,0%	26,7%	26,3%

In table 2, for the Diabetes dataset, the error rate of a single RN was found in last column to be about 25% on the same test set. The RN committee outperforms majority voting and manages to be as accurate as the single RN which was unexpected.

Table 3. Wisconsin dataset results

RN modules	Majority Voting error	RN committee error	single RN error
10	3,7%	3,5%	3,7%
20	3,6%	2,9%	3,2%
30	4,3%	3,6%	3,3%
40	4,4%	3,3%	3,5%
50	4,1%	3,5%	3,5%
60	4,9%	3,5%	3,8%

In table 3, for the Wisconsin dataset, the RN committee again performs better than majority voting and achieves the same error as the single RN, which also marginally overrun in some cases.

Table 4. Vehicle dataset results

RN modules	Majority Voting error	RN committee error	single RN error
5	27,3%	23,4%	21,5%
10	31,3%	25,4%	20,7%
15	33,2%	27,7%	21,1%
20	32,4%	27,3%	20,7%
25	33,6%	28,5%	20,7%
30	37,5%	30,1%	21,1%

In table 4, for the Vehicle dataset the single RN error rate in last column was found to be about 21%. The RN committee performs much better than majority voting and as expected in all cases the error produce where in between majority voting and the single RN error.

Table 5. Glass dataset results

RN modules	Majority Voting error	RN committee error	single RN error
10	38,8%	34,3%	32,8%
12	40,3%	38,8%	33,6%
14	38,8%	37,3%	32,1%
16	39,6%	36,6%	32,8%
18	43,3%	35,8%	34,3%
20	44,8%	37,3%	32,8%

In table 5, for the Glass dataset, again the RN committee performs much better than majority voting. As expected from the highly uneven and un-stratified data distribution across processors in all cases the RN committee error produce where in between majority voting and the single RN error.

Table 6. Wine dataset results

RN modules	Majority Voting error	RN committee error	single RN error
11	3,6%	2,7%	2,4%
12	2,5%	2,5%	2,5%
13	3,1%	3,1%	2,5%
14	3,8%	2,7%	2,9%
15	2,7%	1,8%	1,8%
16	3,6%	2,0%	2,2%
17	3,8%	2,7%	2,7%

In table 6, for the Wine dataset, again the RN committee error results are better than majority voting and once more are comparable to the single RN case. While we run the method 60 times for each dataset, more experiments are needed, and are therefore planed for future research in an extensive collection of benchmark datasets different in record size and feature complexity, together with another training method.

7 Conclusions and future work

For large scale distributed committee machines we consider the challenging case where no local data exchange is possible among the neural network classifiers. Regularization neural networks are used for both the classifiers as well as the combiner committee in an embedded architecture. After the RN committee training finished no RN module will know anything else except its own input local data vectors. The present study proposes a simple method to accomplish such a task. Using the distributed system a mutual validation matrix among them is computed asynchronously. The mapping is done based on classification rates between them. The train set of one becomes the validation set of the other. Then it is possible to exploit this mutual validation matrix to train another high level regularization network as a RN committee combiner for the individual RN modules. Experimental results were supportive, and the proposed privacy preserving RN committee outperforms the majority voting rule in all of the cases.

It must be noted here that as the mutual validation method improves accuracy, the gaining speed is also remarkable, producing a highly scalable system. The complexity of a single RN is about O(N^3). For N>1.000.000, this algorithm is difficult to implement. It is possible for the RN committee machine presented here to assist in splitting the work without significant loss of accuracy. Training with fine-grained modules can be done by globally finding compact dense clusters of data. In the future we will try using a different mutual validation matrix for each class. Since these are asymmetric matrices and might have zeros in the diagonal we plan to resolve this issue by using regularized alternating least squares for the weights training.

Let a general loss function denoted as V(f(x), y, u) where f() the classifier, x the feature vector, y the label and u the parameters vector (weights etc.) of the classifier. The minimization of V with respect to u like in eq. 1 gives the solution of parameter vector u [4]. When two classifiers are compared versus a common separate test set, the comparison is made on their outputs, so their distance measure d(i,j) is usually dependent on their pair of parameter vectors u_1 and u_2, meaning the outcome d(i,j) is biased from their joint biases. The proposed mutual validation matrix method is independent of the joint parameter vectors u_1 and u_2. In our case an asymmetric measure s(i,j) depends only on parameter vector u_1 of classifier i and thus is independent from the bias and variance of the classifier j. So it can be used as uncorrelated distance measure estimation for conventional ensemble training.

Unlike the on-line neural network training, or gradient descent methods, the training phase of a regularization network is always off-line, using kernel methods, and thus the stable solution is restricted to solving a linear system of the form $(K + \lambda I)w = y$. Thus with or without the privacy preserving constrain, the static training of a meta-learner RN committee which consists of any other type of classifier modules (SVM, RBF, MLP etc.) requires a coarse-grained high level kernel matrix. For example constrained regression training uses a covariance matrix usually computed from the average errors of classifiers to find module weights. In the future we plan to present extensive experiments that directly compare the proposed method performance with ensembles of RNs trained via bagging, constrained regression and stacking.

8 References

1. Tresp, V.: Committee Machines. Chapter in Handbook of neural network signal processing, Hu, Y. H., and Hwang, J.N., eds., CRC Press LLC. (2002).
2. Drucker, H.: Fast Committee Machines for Regression and Classification. KDD-97 Proceedings, (1997).
3. Girosi, F., Jones, M., Poggio, T.: Regularization theory and neural networks architectures. Neural Computation 7, 219-269 (1995).
4. Evgeniou, T., Pontil M., Poggio, T.: Regularization Networks and Support Vector Machines. Advances in Computational Mathematics 13, 1–50 (2000).
5. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Notices of the American Mathematical Society 50 (5), 537–544 (2003).
6. Bishop, C. M.: Neural Networks for Pattern Recognition. Oxford: Oxford University Press, (1995).
7. Jain, A. K., Duin R. P. W., Mao, J.: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4–37 (2000).
8. Wang, L., Fu, X.: Data Mining with Computational Intelligence. Springer-Verlag (2005).
9. Bottou, L., Chapelle, O., DeCoste, D., Weston, J.: Large Scale Kernel Machines. Neural Information Processing Series, MIT Press, Cambridge, MA. (2007).
10. Kashima, H., Ide, T., Kato, T., Sugiyama, M.: Recent Advances and Trends in Large-scale Kernel Methods. IEICE Transactions on Information and systems E92-D, (7), 1338-1353 (2009).
11. Prodromidis, A., Chan, P.: Meta-learning in a distributed data mining system: Issues and approaches. In Proceedings of the Fourteenth International Conference on Machine Learning, 211-218, (1998).
12. Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. and Zhu, M.: Tools for Privacy Preserving Distributed Data Mining. ACM SIGKDD Explorations, 4 (2), 1-7 (2003).
13. Kantarcioglu M., Vaidya J.: Privacy-Preserving Naive Bayes Classifier for Horizontally Partitioned Data. IEEE Workshop on Privacy-Preserving Data Mining, (2003).
14. Yi, X., Zhang Y.: Privacy-preserving naïve Bayes classification on distributed data via semi-trusted mixers. Information Systems 34(3), 371–380 (2009).
15. Yu, H., Jiang, X., Vaidya J.: Privacy-Preserving SVM using nonlinear Kernels on Horizontally Partitioned Data. SAC Conference, (2006).
16. Xiong, L., Chitti, S., Liu, L.: k nearest neighbour classification across multiple private databases. In Proceedings of the ACM Fifteenth Conference on Information and Knowledge Management, November 5-11, (2006).
17. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001 (1990).
18. Perrone, M.P., Cooper, L.N.: When networks disagree: ensemble method for neural networks. in Neural Networks for Speech and Image Processing, R.J. Mammone, Ed., Chapman & Hall, Boca Raton, FL, (1993).
19. Krogh, A., Vedelsby, J.: Neural networks ensembles, cross validation and active learning. in Advances in Neural Information Processing Systems 7, MIT Press, Cambridge, (1995).
20. Wilson, G.: Parallel Programming for Scientists and Engineers, MIT Press: Cambridge (1995).