Random Forest-like strategies for Neural Network Ensembles Construction

Rafael Namías and Pablo M. Granitto
IFIR – CIFASIS, CONICET/UNR
Bv. 27 de Febrero 210 Bis, 2000 Rosario, Argentina
granitto@ifir.edu.ar

Abstract

Ensemble methods show improved generalization capabilities that outperform those of single learners. It is generally accepted that, for aggregation to be effective, the individual learners must be as accurate and diverse as possible. An important problem in ensemble learning is then how to find a good balance between these two conflicting conditions. For tree-based methods a successful strategy was introduced by Breiman with the Random-Forest algorithm. In this work we introduce new methods for neural network ensemble construction that follow Random-Forest-like strategies to construct ensembles. Using several real and artificial regression problems, we compare our new methods with the more typical Bagging algorithm and with three state-of-the-art regression methods. We find that our algorithms produce very good results on several datasets. Some evidence suggest that our new methods work better on problems with several redundant or noisy inputs.

Keywords: Machine Learning; Ensemble Methods; Neural Networks; Random Forest

1 Introduction

Over the last decade ensemble methods have been on the focus of machine learning research[10, 17]. The base of these procedures is the intuitive idea that by combining the outputs of several individual predictors one might improve on the performance of a single generic one. The so-called bias/variance dilemma [6] provides formal support to the success of these strategies. According to these ideas, good ensemble members must be both accurate and diverse, which poses the problem of generating a set of predictors with reasonably good individual performances and independently distributed predictions for the test points. As these are two opposite conditions, good ensemble methods achieve a compromise between them. Typical examples are bagging [2] and boosting [5].

Diverse individual predictors can be obtained in several ways. Bagging and boosting strategies are based on learning from different adequately-chosen subsets of the data set. Other methods try to generate diversity by adding small perturbations to the data at hand. For example, Rodriguez et al. [12] use partial PCA decompositions at each node of a CART Tree. Breiman [4] randomizes the class labels of a small sub-sample of the data when growing each tree and Martinez et al. [11] perform the same procedure for neural networks ensembles. Another successful strategy is to modify slightly the internal learning structure of a given algorithm in order to gain in diversity. For example, Geurts et al. [7] grow decision trees using randomly selected variables and split points. Clearly, the most successful method of this class is the Random Forest (RF) algorithm [3], introduced by Leo Breiman. In RF, like in bagging, several CART trees are grown on bootstrap samples of the original dataset. But, when growing each tree, only a small random subset of features is considered at each node. Doing this, at each step the algorithm minimizes a cost function only in a randomly selected subspace of the full hypothesis space.

Several ensemble techniques have been recently applied to artificial neural networks (ANN) [8, 14, 16]. As the diversity of ANN comes naturally from the training process randomness and from the intrinsic non-identifiability of the model, it is difficult to improve over simple strategies like using several networks trained on the same data or plain bagging. For classification problems boosting of ANN outperforms other ensemble methods in several cases [16, 13] but for regression problems several methods show similar results [8]. Aiming at increasing the diversity in ANN ensembles, in this work we introduce new methods for neural network ensemble construction that follow Random-Forest-like strategies, i.e., that minimize the ANN cost function on randomly selected subspaces. We test the proposed algorithms on regression problems, using real and artificial benchmark datasets and time series.

This paper is organized as follows. In the next section we recall the bias/variance dilemma. Next, we introduce our new ensemble construction strategies. In Section 4 we describe the experimental settings used and in Section 5 we show and discuss the empirical results of our new methods. Finally, in Section 6 we draw some conclusions and discuss future lines of research.

2 Bias and Variance

In this section we will briefly recall the bias/variance decomposition of the generalization error [6] (which is the theoretical base of ensemble methods) following [9]. Let us consider a set of N noisy data pairs $D=\{(t_i, \mathbf{x}_i),\, i=1,N\}$, where the vectors $\mathbf{x}_i$ of predictor variables are obtained from some distribution $P(\mathbf{x})$ and the regression targets t_i are generated according to

$$t_i = f(\mathbf{x}_i) + \varepsilon_i. \tag{1}$$

Here f is the true regression and ε is random noise with zero mean. If we estimate f using D obtaining a model f_D, the (quadratic) generalization error on a test point $(t, \mathbf{x})$ averaged over all possible realizations of the data set D (with respect to P and noise ε) can be decomposed as:

$$\mathrm{E}[(t - f_D(\mathbf{x}))^2|D] = \mathrm{E}[\varepsilon^2|\varepsilon] + (\mathrm{E}[f_D(\mathbf{x})|D] - f(\mathbf{x}))^2 + \mathrm{E}[(f_D(\mathbf{x}) - \mathrm{E}[f_D(\mathbf{x})|D])^2|D] \tag{2}$$

The first term on the right-hand side is simply the noise variance σ_ε^2; the second and third terms are, respectively, the squared bias and variance of the estimation method. For a single model f_D we can interpret this equation by saying that a good method should be no biased and have as little variance as possible between different realizations.

If we rewrite the error decomposition in the form:

$$\mathrm{E}[(t - \mathrm{E}[f_D(\mathbf{x})|D])^2|D] = \mathrm{Bias}^2 + \sigma_\varepsilon^2 = \mathrm{MeanError} - \mathrm{Variance}, \tag{3}$$

we can reinterpret this equation in the following way: using the average $\mathrm{E}[f_D|D]$ as estimator, the generalization error can be reduced if we are able to produce fairly accurate models f_D (small MeanError[1]) while, at the same time, allowing them to produce the most diverse predictions at

[1] In this work we use the terms Accuracy and MeanError with the same meaning. In other cases Accuracy is defined as –MeanError or in another mathematical form that has a derivative opposite to MeanError's one.

every point (large Variance). Of course, there is a trade-off between these two conditions and several previous works [8, 14] discussed how to find a good compromise between mean error and diversity on ANN ensembles.

3 Learning in random subspaces

RF, as was discussed in the Introduction, is one of the most successful tree-based ensemble methods. RF combines two different sources of diversity: i) each tree in the ensemble is grown on a bootstrap sample of the original dataset and ii) when growing each tree, only a small random subset of features is considered at each node. By this second condition the learning algorithm is restricted to minimize its cost function only in randomly selected subspaces of the original hypothesis space. According to Breiman, this procedure produces less correlated trees while keeping a low mean error.

We can incorporate this last source of diversity to ANN ensembles with simple modifications of the training procedure. All practical ANN learning methods are based on an iterative minimization of a cost function over the vector space of possible weight values [1]. In all cases we can easily restrict the minimization procedure to random subspaces by the following procedure:
i) select at random a subset of weights.
ii) iterate a few cycles the learning algorithm but limiting it to change only the selected weights.
iii) iterate steps i) and ii) until a stopping criterion is reached.
Of course, different stopping criteria or random selection strategies can be implemented, leading to slightly different versions of the method. It worth mention that at step ii) all weights are available to the training algorithm to compute ANN outputs but it is limited to train (change) only the selected subset of weights.

This work is limited to model regression problems using ensembles of ANNs with a single hidden layer (with sigmoid activation functions) and a single output unit (with linear activation). For this particular setting we choose to keep all weights connecting the output unit to the hidden layer[2] and to limit the random selection to weights connecting the hidden layer to the input units using the following two strategies:
i) simply select at random a fraction F of all weights and pass them to the learning method. We call this strategy weight selection (WS),
ii) select a fraction F of input units and pass all weights coming from these units to the learning method. We call this strategy input selection (IS).
We also consider two different stopping criteria for the training of individual ANNs. In the first case, called optimal training (OT), we use the out-of-bag data as a validation set in order to monitor the performance of each ANN on unseen data and avoid overfitting. This is the most usual strategy for ANN or tree based ensembles. It is well known [3, 8] that some degree of overfitting of the individual members can be of benefit for the ensemble performance. We thus use a second criterion by which we train all ANNs a fixed number of epochs, long enough as to be sure that we overfit the training data. We call this second criterion full training (FT). Combining the two selection and stopping criteria plus different values of the fraction F of selected weights we produce diverse methods with different compromises between accuracy and variance.

[2] As all ANNs have only one output unit, selecting a subset of connections for that unit can reduce the effective complexity of the model and produce an accuracy decrease without the corresponding variance increase.

4 Experimental Settings

We evaluate the algorithms described in the previous section by applying them to several benchmark databases: the synthetic Friedman #1 and #2, the real-world Boston Housing and Ozone and two times series, Sunspots and Ikeda. In the following we give brief descriptions of the databases and details on the experimental settings.

4.1 Datasets

Friedman #1

The Friedman #1 synthetic data set corresponds to vectors with 10 input and one output variables generated according to

$$t = 10\sin(x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon,$$

where ε is Gaussian noise ($N(\mu = 0, \sigma = 1)$) and $x_1, \ldots x_{10}$ are uniformly distributed over the interval $[0, 1]$. Notice that $x_6, \ldots x_{10}$ do not enter in the definition of t and are only included to add input noise. We generate 1100 sample vectors and consider ANNs with 10:10:1 architectures.

Friedman #2

Friedman #2 has four independent variables and the target data are generated according to

$$y = x_1^2 + \sqrt{x_2 x_3 - (x_2 x_4)^{-2}} + \varepsilon,$$

where the zero-mean, normal noise is adjusted to give a noise-to-signal power ratio 1:3. The variables x_i are uniformly distributed in the ranges

$$0 < x_1 < 100, \quad 20 < \frac{x_2}{2\pi} < 280, \quad 0 < x_3 < 1, \quad 1 < x_4 < 11.$$

We generate 1100 patterns and consider 4:10:1 ANNs.

Boston Housing

This data set, from the UCI machine learning repository, consists of 506 training vectors with 11 input variables and one target output. The inputs are mainly socioeconomic information from census tracts on the greater Boston area and the output is the median housing price in the tract. We select an 11:8:1 architecture for ANNs.

Ozone

The Ozone data correspond to meteorological information (humidity, temperature, etc.) related to the maximum daily ozone at a location in Los Angeles area. Removing missing values one is left with 330 training vectors, containing 8 inputs and one target output in each one. The data set can be downloaded University of California at Berkeley. We select a 8:8:1 architecture in this case.

Method	Fried#1	Fried#2	Ozone	Boston	Ikeda	SSP
Bagging	0.30	0.12	0.25	0.12	0.29	0.12
WS-90%	0.89	-	1.02	0.99	0.99	-
WS-80%	0.94	1.03	0.99	1.02	1.00	-
WS-70%	0.97	1.02	1.04	1.04	1.07	1.03
WS-60%	1.00	1.02	1.01	0.97	1.04	1.03
WS-50%	1.03	1.04	1.02	0.95	1.05	1.02
WS-40%	-	-	-	-	-	1.03
IS-90%	0.99	-	-	0.94	1.00	-
IS-80%	0.98	1.01	1.00	0.98	1.00	-
IS-70%	0.97	1.00	1.02	0.91	0.97	0.99
IS-60%	0.93	0.99	0.98	0.98	1.02	0.99
IS-50%	0.90	0.99	0.98	1.05	1.04	0.99
IS-40%	-	-	-	-	-	0.98
IS-30%	-	-	-	-	-	0.97

Table 1: Experimental results for the Optimal Training stopping criterion. Bagging results are given in NMSE units. Results of all other methods are in relative units to the corresponding bagging value (i.e., a value lower than one means increasing in performance over bagging). Each row corresponds to a different selection strategy and fraction of selected weights.

Ikeda

The Ikeda laser map, which describes instabilities in the transmitted light by a ring cavity system, is given by the real part of the complex iterates

$$z_{n+1} = 1 + 0.9 z_n \exp\left[0.4i - \frac{6i}{(1+|z_n|^2)}\right].$$

We generate 605 iterates and create 600 vectors using as inputs the last five values of the time series and as output the corresponding next value. We consider 5:8:1 ANNs.

Sunspots

The sunspots (SSP) time series is one of the most used benchmarks in time series prediction. It is the record of the yearly average of the number of sunspots (dark blotches on the sun mainly caused by magnetic activity) since 1700 to 1999. We generate 287 vectors using as inputs the last 12 values of the time series and as output the corresponding next value. We consider 12:6:1 ANNs for this problem.

4.2 Evaluation methods

For all six datasets we use ensembles with 50 ANNs. We selected this number of networks after checking in all cases that there are practically no performance improvements with bigger ensembles. Each ANN is trained with the standard back-propagation algorithm with momentum[1]. Training parameters (momentum, learning rate and # of epochs) were selected by internal cross validation but without a in-depth search for optimal values, because we are mainly interested in the relative performance of different ensemble methods.

Method	Fried#1	Fried#2	Ozone	Boston	Ikeda	Ssp
Bagg-FT	0.93	1.43	1.10	0.97	0.99	1.02
WS-70%	0.85	1.40	1.14	0.91	1.00	0.96
WS-60%	0.90	1.33	1.15	0.90	1.00	0.96
WS-50%	0.94	1.39	1.13	0.91	1.01	0.94
WS-40%	0.97	1.39	1.17	0.90	1.06	0.95
WS-30%	1.00	-	1.15	-	-	0.94
IS-70%	0.88	-	1.12	0.90	1.00	0.95
IS-60%	0.84	1.42	1.13	0.84	0.99	0.93
IS-50%	0.83	1.35	1.17	0.88	1.00	0.92
IS-40%	0.79	1.37	1.10	0.91	1.00	0.93
IS-30%	0.75	1.35	1.11	0.91	1.00	0.91

Table 2: Experimental results for the Full Training stopping criterion. All results are in relative units to the corresponding bagging value (see Table 1). Each row corresponds to a different selection strategy and fraction of selected weights.

All the results given in the next section are averages over 100 runs of each method. We repeated 10 times a 10-folds cross validation procedure, using alternatively one fold as test set and the remaining nine as training set. For all the methods under evaluation we use exactly the same 100 partitions in training and test sets.

The results quoted below are given in terms of the normalized mean-squared test error:

$$NMSE_T = \frac{MSE_T}{\sigma^2_D}, \tag{4}$$

defined as the mean-squared error on the test set T divided by the variance of the total data set D. For easy of interpretation, the results of the baseline method (bagging of optimally trained ANNs) are given in these units and the results of the other methods are given relative to these values. For example, the value 0.89 corresponding to Fried#1, WS-90% in Table 1 means that the WS-90% strategy gives more than a 10% decrease in test set prediction error over plain bagging.

5 Experimental results

We start our analysis evaluating the more typical OT stopping criterion. For both WS and IS selection strategies we use several values of the fraction F of selected weights in the 30–90% range. In Table 1 we show the corresponding results, including for comparison the results of bagging of optimally trained ANNs. Note that the only difference between bagging and the WS and IS strategies is the limitation on the last two methods to train only the selected subset of weights. For the WS strategy the results are poor. Only for Friedman#1 and Boston there are improvements over bagging for some values of F. In all other cases the OT–WS strategy produce worse results than bagging. On the other hand, the IS strategy outperforms WS in all six datasets. For this method, OT–IS, there are consistently better than bagging results for Friedman#1, Boston and SSP, and similar to bagging prediction errors for the other three

Dataset	Fried#1		Boston		Ssp	
Method	Acc	Var	Acc	Var	Acc	Var
Bagging	0.46	0.16	0.20	0.086	0.156	0.033
Bagg-FT	1.75	3.32	1.41	2.17	1.02	0.99
WS-70%	1.79	3.51	1.29	1.83	1.61	4.05
WS-60%	1.82	3.51	1.27	1.77	1.56	3.83
WS-50%	1.88	3.65	1.23	1.67	1.49	3.57
WS-40%	1.96	3.83	1.21	1.61	1.43	3.21
WS-30%	-	-	-	-	1.34	2.85
IS-70%	1.73	3.36	1.41	2.10	1.67	4.40
IS-60%	1.72	3.39	1.34	2.01	1.63	4.20
IS-50%	1.70	3.38	1.32	1.93	1.48	3.60
IS-40%	1.66	3.33	1.33	1.89	1.43	3.31
IS-30%	1.64	3.34	1.27	1.82	1.34	2.96

Table 3: Mean Error and Variance for three datasets (Fried#1, Boston and SSP) with similar behavior. Results are given in units relative to the corresponding bagging results. Columns labeled Acc show accuracy values (lower values are better) and columns labeled Var show variance values (bigger values are better).

problems. Analysing the dependence on the F value there is not a clear pattern, in some cases there are improvements using small subspaces and in other cases the opposite is true.

We repeat the analysis for the FT stopping criterion. In table 2 we show the new results, but including this time the results of bagging of fully trained ANNs (bagg-FT). Again, for the FT stopping criterion, the only difference between bagg-FT and the WS and IS strategies is that the last two methods train only a selected subset of weights. The bagg-FT results are similar to plain bagging for three datasets, worst for Fried#2 and Ozono and better only for Fried#1. IS is almost always better than WS, which is consistent with the OT results, suggesting that the IS strategy is more efficient. Comparing with bagg-FT there are two different behaviors. For Fried#1, Boston and SSP there are now clear improvements over bagging and bagg-FT. On the other side, for Fried#2, Ozono and Ikeda the results are similar or slightly worst than bagg-FT, and clearly worst than bagging for the first two. As in the OT case, there is not a clear pattern for the dependence on the F value, which seems to be problem-dependent.

Comparing the OT and FT stopping criteria, both show improvements over bagging on the same three datasets and some decrease in performance on the other three, but the FT strategy produces bigger differences with bagging in all cases.

5.1 Accuracy vs. Diversity

In order to gain some insight on the behavior of the new methods we also estimate the accuracy and diversity components of the prediction error according to section 2, equations 2 and 3. We limit the analysis to the IS strategy, which produces the bigger improvements over bagging.

In Tables 3 and 4 we present the corresponding results; for easier comparison, we again give them normalized by the mean accuracy and diversity of plain bagging and give bagging results in NMSE units. In Table 3 we present the results for the three datasets that clearly improve on

Dataset	Fried#2		Ozone		Ikeda	
Method	Acc	Var	Acc	Var	Acc	Var
Bagging	0.16	0.035	0.31	0.058	0.716	0.43
Bagg-FT	3.07	8.87	2.05	6.20	2.05	2.74
WS-70%	2.43	6.08	1.85	4.96	1.95	2.56
WS-60%	2.39	6.14	1.82	4.76	2.03	2.71
WS-50%	2.56	6.68	1.75	4.46	1.99	2.62
WS-40%	2.48	6.34	1.58	3.38	2.03	2.71
WS-30%	-	-	1.48	2.97	-	-
IS-70%	2.80	7.67	1.96	5.68	2.06	2.75
IS-60%	2.34	5.80	1.89	5.20	2.08	2.78
IS-50%	2.33	5.69	1.81	4.63	2.00	2.65
IS-40%	2.33	5.82	1.79	4.79	1.99	2.63
IS-30%	-	-	1.67	4.10	2.00	2.66

Table 4: Mean Error and Variance for three datasets (Fried#2, Ozone and Ikeda) with similar behavior. Results are given in units relative to the corresponding bagging results. Columns labeled Acc show accuracy values (lower values are better) and columns labeled Var show variance values (bigger values are better).

bagging, Fried#1, Boston and SSP, and left the other three datasets for Table 4. The bagg-FT values in both tables show that the FT strategy produces a lost in accuracy linked with an increase in variance, relative to bagging values, in all but the SSP dataset. Comparing both groups of datasets, the first one (Table 3) has lower increases in both accuracy and variance over bagging.

The accuracy and variance values for the WS and IS strategies are not easy to analyze. For SSP there is, as expected, a four times increase in variance coupled with a moderate loss in accuracy (bigger Acc values), given as result a reduced prediction error (see Table 2). But the results for Fried#1 and Boston are unexpected. In both datasets the better performance is associated to accuracy values better than bagg-FT together with increases in variance similar to bagg-FT. For all datasets in Table 4 our subspace strategies also produce better than bagg-FT accuracies, but coupled in this case with a reduction in variance, which seems to be the cause of the lost in prediction capabilities.

5.2 Adding noisy inputs

The three datasets that show improvements in prediction error over bagging and bagg–FT have some very noisy or irrelevant inputs. The Fried#1 datasets has 5 white noise inputs. Our embedding of the SSP datasets uses the last 12 values of the series, but is know that a non-uniform embedding with 3 values gives optimal results. The inputs in the Boston problem consist of very noisy socioeconomic information from census tracts. The other three datasets have fewer inputs and all relevant. To check if this can be the origin of the different behavior of the two groups we conducted a small experiment adding white-noise inputs to the Fried#2 dataset. We produced two new datasets, one with 5 added noisy inputs and a second one with 15 more noisy inputs. We show the new results in Table 5. The addition of useless inputs produces a better performance of our FT subspace methods relative to that of bagging and bagg–FT. The

	Plain			+5 Features			+20 Features		
Method	Err	Acc	Var	Err	Acc	Var	Err	Acc	Var
Bagging	0.12	0.16	0.035	0.19	0.60	0.41	0.27	0.69	0.42
Bagg-FT	1.43	3.08	8.87	0.96	1.00	1.01	0.99	0.97	0.96
WS-FT-50%	1.39	2.56	6.68	0.98	1.02	1.02	1.04	1.04	1.03
IS-FT-50%	1.35	2.33	5.69	1.03	1.25	1.34	0.98	0.99	0.99

Table 5: Prediction error (Err), Accuracy (Acc) and variance (Var) for the Friedman#2 dataset with added white-noise input features. Results are given in units relative to the corresponding bagging results.

results are equivalent for 5 or 20 added features, but for 20 features the IS-FT strategy gives the best result of all methods.

5.3 Comparison with other methods

As a final investigation on our random subspace methods we compare them with three other state-of-the-art regression methods. We selected two tree-based ensemble methods, bagging and Random Forest [2, 3] and Support Vector Machines (SVM) [15] with a gaussian kernel. We selected the FT-IS-50% method as a good representative of our new methods. To have a fair comparison we use exactly the same 100 partitions in train/test set. We use 1000 trees for both Bagging and RF and set all other parameters to the default values given by Breiman. For SVM we selected the C and γ parameters using internal cross-validation on each train set.

The corresponding results are shown in Table 6. On Fried#1, Boston, Ikeda and SSP datasets the FT–IS–50% method gives the best results. On the other two datasets, tree-bagging wins in one case and RF in the other. But in almost all cases tree-bagging, RF and SVM have bigger than one results, showing that in fact it is really difficult to improve over the results of plain bagging of optimally trained ANNs.

6 Conclusions

In this work we introduced new methods for neural network ensemble construction that follow Random-Forest-like strategies in order to increase the diversity of the members. We selected two strategies for the random selection of weights to be trained, the WS strategy that simply selects

Method	Fried#1	Fried#2	Ozone	Boston	Ikeda	Ssp
Bagging (trees)	1.13	1.22	1.09	1.11	1.47	1.66
RF (trees)	1.26	2.53	1.00	1.09	1.66	2.02
SVM (gaussian)	0.85	1.39	1.14	1.94	1.21	1.32
FT-IS-50%	0.83	1.35	1.17	0.88	1.00	0.92

Table 6: Prediction error comparison with other state-of-the-art methods. Results are given in units relative to the corresponding bagging results.

at random a fraction F of all weights and the IS strategy that selects a fraction F of input units and all the weights coming from these units. We also consider two different stopping criteria for the training of individual ANNs, the typical optimal training (OT) and the full training (FT) by which we train all ANNs a fixed number of epochs, long enough as to overfit the training data. We evaluated the combination of the two selection and stopping criteria plus different values of the fraction F of selected weights on six real-world and artificial regression problems. We found that the IS strategy usually outperforms the WS one, and that the FT stopping method gives better than OT results for three datasets where all subspace strategies outperform bagging. On the other three datasets both stopping criteria works no better than bagging and in particular FT gives the worst results.

The analysis of the accuracy and variance values suggest that in most cases the subspace strategies do not produce an increase in variance over fully trained bagging ensembles. Instead, they seem to produce better prediction errors by a combination of an increased accuracy with similar variance than bagging. This result is very interesting and requires further investigations.

We have also evaluated the addition of noisy inputs to the Fried#2 dataset. The results of that experiment supports the idea that our subspace methods perform better in problems with several noisy or irrelevant inputs. This fact suggests an important application field, the calibration of spectrometric instrument in chemometrics.

Finally, we have also performed a comparison with other three state-of-the-art regression methods. The FT-IS strategy was the only method of the four evaluated capable of clearly outperforming bagging of ANNs in several datasets.

As future work we are also considering extending the proposed methods to classification problems and to evaluate other random selection strategies.

Acknowledgments

We acknowledge support for this project from the ANPCyT (grant PICT 2003 11-15132).

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, London, 1995.

[2] L. Breiman. Bagging predictors. *Machine Learning* 24:123-140, 1996.

[3] L. Breiman. Random forests. *Machine Learning* 45:5-32, 2001.

[4] L. Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning* 40:229-242, 2000.

[5] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23-37, Springer Verlag, 1995.

[6] S. Geman, E. Bienenstock and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation* 4:1-58, 1992.

[7] P. Geurts, D. Ernst and L. Wehenkel. Extremely randomized trees. *Machine Learning* 63:342, 2006

[8] P. M. Granitto, P. F. Verdes and H. A. Ceccatto. Neural Networks Ensembles: Evaluation of Aggregation algorithms. *Artificial Intelligence* 163:139-162, 2005.

[9] P. M. Granitto, P. F. Verdes, H. D. Navone and H. A. Ceccatto. A Late-stopping Method for Optimal Aggregation of Neural Networks. *Internation Journal of Neural Networks* 11:305-310, 2001.

[10] L. I. Kuncheva *Combining Pattern Classifiers*. Wiley-Interscience, New Jersey, 2004.

[11] G. Martinez-Muñoz, A. Sanchez-Martinez, D. Hernandez-Lobato and A. Suarez. Building Ensembles of Neural Networks with Class-Switching. Proceedings of the ICANN 2006, Part I, LNCS 4131, 178187, 2006.

[12] J J Rodriguez, L. I. Kuncheva and C. J. Alonso. Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1619-1630, 2006.

[13] D. Opitz and R. Maclin. Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, 11:169-198, 1999.

[14] B. Rosen. Ensemble learning using decorrelated neural networks. *Connection Science*. Special Issue on Combining Artificial Neural Nets: Ensemble Approaches 8(3&4):373-384, 1996.

[15] B. Scholkopf and A. J. Smola. *Learning with Kernels* MIT Press, Cambridge, 2002.

[16] H. Schwenk and Y. Bengio. Boosting neural networks. *Neural Computation* 12:1869-1887, 2000.

[17] A. J. C. Sharkey, editor. *Combining Artificial Neural Nets*. Springer-Verlag, London, 1999.