Model Generation of Neural Network Ensembles Using Two-Level Cross-Validation

S. Vasupongayya, R.S. Renner, and B.A. Juliano

Department of Computer Science
California State University, Chico
Chico, CA 95929-0410
{sang, renner, juliano}@ecst.csuchico.edu

Abstract. This research investigates cross-validation techniques for performing neural network ensemble generation and performance evaluation. The chosen framework is the Neural Network Ensemble Simulator (NNES). Ensembles of classifiers are generated using *level-one cross-validation*. Extensive modeling is performed and evaluated using *level-two cross-validation*. NNES 4.0 automatically generates unique data sets for each student and each ensemble within a model. The results of this study confirm that *level-one cross-validation* improves ensemble model generation. Results also demonstrate the value of *level-two cross-validation* as a mechanism for measuring the true performance of a given model.

1 Introduction

In a traditional neural network system, a model is represented by an individual network, one that has been trained on a single data set for a specific domain. Such a system can be replaced by an "ensemble" [3][5][6], a system model composed of multiple individual neural networks. In this study, the process of creating an ensemble consists of training each network in the ensemble individually using a unique training set, validating each network using a unique validation set, and combining all networks to form an ensemble using the *weighted contribution* combination method [10].

A neural network model, represented by either an individual neural network or an ensemble, is considered "good" if it is able to generalize over the entire domain and correctly predict or classify unseen data [14]. In order to generate good ensembles of unique neural networks, a sufficient amount of available data is needed for the training and validation processes. In reality, the available data are limited, so it is important to employ optimal usage of these data. Conventionally, researchers have worked around the limited data to achieve unique networks by using one or more of four methods: (1) changing the initial topology, (2) changing the tuning parameters, (3) using multiple learning algorithms, or (4) using different training data. According to Amari [1], with a fixed data set, the first three of these methods may lead to the *overfitting* or *overtraining* problem, because the training data are potentially biased.

Cross-validation is one of several techniques of using different training data to achieve unique networks [7][8]. *Cross-validation* rotates the training data and thus reduces the *bias* that leads to *overfitting*. There are two levels of *cross-validation* proposed in this study. The first one is *level-one cross-validation* (CV1), which

V.N. Alexandrov et al. (Eds.): ICCS 2001, LNCS 2074, pp. 943–951, 2001.

potentially achieves unique networks by rotating the training and validation data. This research claims that CV1 will make *overfitting* to the entire training data set less likely.

Based on this research, CV1 as a method for *cross-validation* not only reduces the *overfitting* problem, it also eliminates the *bias* problem caused by the location of the test set in the sample data. Since it is commonly understood that a good neural network should be able to generalize over the entire domain and correctly classify unseen data, using one particular test set can lead to a misrepresentation in the performance measurement. The second level of *cross-validation* proposed in this study is called *level-two cross-validation* (CV2). CV2 eliminates the bias by grabbing a new test set each time a new ensemble is generated. The overall performance of the model is represented by the average performance over all ensembles in a given model.

2 Two-Level Cross-Validation

2.1. Level-Two Cross-Validation

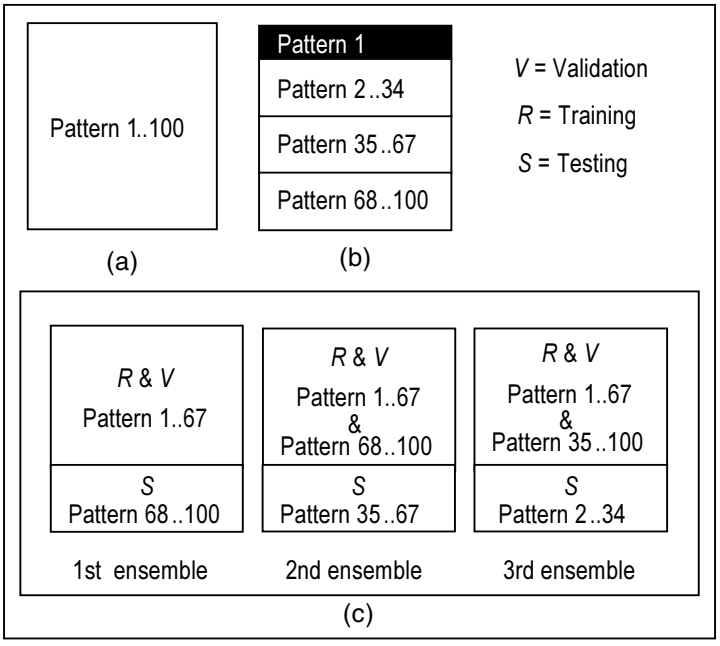

Fig. 1. An example of how the CV2 process rotates the test set for P=100 data patterns and K_2=3 ensembles to generate ensemble data sets: (a) An original example data file with 100 data patterns; (b) Data are divided into K_2=3 groups of equal size with 1 data pattern left; (c) The resulting ensemble data sets

By using the CV2 ensemble creation procedure, all the available P data patterns are divided into K_2 non-overlapping groups, where K_2 is the number of ensembles. Each ensemble gets a different test set consisting of P/K_2 data patterns and the rest of the available data are used as the training and validation set. If the division is uneven, the left over data patterns are always included in the training set for all ensembles.

Figure 1 illustrates the CV2 process for K_2=3 ensembles and P=100 data patterns. The test set for the first ensemble consists of 33 data patterns, numbered 68 to 100 in the original file. The remaining data, numbered 1 to 67 in the original file, are used as the training and validation set of the first ensemble. This way all data are used and each ensemble gets a test set of equal size.

2.2. Level-One Cross-Validation

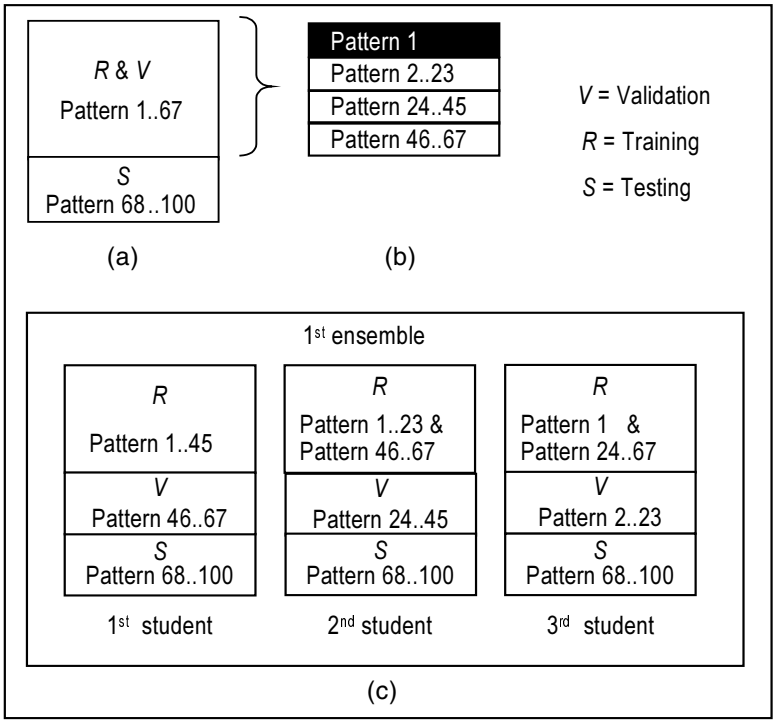

Fig. 2. An example of how the CV1 process rotates the validation set for P=67 data patterns and K_1=3 students to generate student data sets: (a) The data set for the first ensemble from Figure 1(c); (b) Training and validation data is divided into K_1=3 groups of equal size with 1 data pattern left; (c) The resulting student data sets

After the CV2 process creates each data set, the CV1 creation procedure is called for each K_2 ensemble data set. CV1 divides all the $P'=P-P/K_2$ training and validation data in a given data set into K_1 non-overlapping groups, where K_1 is the

number of networks (students) in an ensemble. Each student is assigned one of the K_1 groups for its validation set. This group provides unique data, with a set size of P'/K_1, to validate each network. The remaining K_1-1 groups of the available data are used as the training set. If the division is uneven the extra patterns are assigned to the training set for all students. Each student gets a non-overlapping validation set, while the training set is overlapping.

Figure 2 illustrates the CV1 process for a training and validation set with $P'=P-P/K_2=100-33=67$ data patterns and $K_1=3$ students. The data set in Figure 2(a) is the data set for the first ensemble given in Figure 1(c). In this example, all $P'=67$ data patterns reserved for the training and validation set are divided into three groups, with one pattern remaining. The validation set for the first student consists of $P'/K_1=67/3=22$ data patterns, numbered 46 to 67 in the original data file from Figure 1(a). All the remaining data patterns, numbered 1 to 45 in the original data file, are used as the training set. In this way, all data are used and each student gets a validation set of equal size.

3 Implementation

The pseudo-code for ensemble creation using CV1 and CV2, within the NNES 4.0:

1. Get the number of ensembles (K_2) and the number of students (K_1)
2. Initialize all parameters using the NNES default values
3. Call the CV2 creation process with K_2
4. Call the CV1 creation process with K_1
5. Generate a network
6. Repeat steps 4-5 until all K_1 networks have been generated
7. Combine all networks to form an ensemble
8. Compute the ensemble performance
9. Repeat steps 2-8 until all K_2 ensembles have been generated and tested

The pseudo-code for the CV2 ensemble creation process:

1. Read in the P data patterns from the data file.
2. Define the test set size as $T = \lfloor P/K_2 \rfloor$.
3. For each *enSeq* where $0 \le enSeq \le K_2$-1,
   a. Use data pattern (P-$T\times enSeq$)-T+1 to data pattern (P-$T\times enSeq$) to form a test set for ensemble number ($enSeq$+1).
   b. Use the remaining $P' = P$-T data patterns to generate the training and validation sets.

The pseudo-code for the CV1 student creation process:

1. Define the validation set size as $V = \lfloor P'/K_1 \rfloor$.
2. For each *stSeq* where $0 \le stSeq \le K_1$-1,

a. Use data pattern (P'-$V\times stSeq$)-V+1 to data pattern (P'-$V\times stSeq$) and reserve them for the validation set for student number ($stSeq$+1).

b. Use the remaining P'-V data patterns in the training set.

4 The Model Considered

The Neural Network Ensemble Simulator (NNES) [10][13] is chosen as the development platform for CV1 and CV2. To support this research, NNES version 3.0 has been updated to version 4.0, which supports CV1 and CV2. Within the NNES 4.0

framework, the *cascade-correlation* learning algorithm is selected to create ensemble models [2]. The *weighted contribution* combination method is selected to combine networks to form an ensemble classifier [10]. The number (K_1) of networks desired for a given ensemble and the number of ensembles (K_2) for a given simulation model are entered by the user. Once the aforementioned parameters are entered, NNES default values are accepted for all remaining parameters.

Results presented in this research are generated from experimentation utilizing the *diabetes* data set from the PROBEN1 repository [9]. The PROBEN1 *diabetes* data set consists of three permutations of the same data set labeled *diabetes1*, *diabetes2* and *diabetes3*. For purposes of comparison to earlier experiments with NNES 3.0 [10][11][12], *diabetes1* is selected as the base data set. *Diabetes1* contains 576 data patterns for training and validation and an additional 192 for testing, for a total of 768 patterns.

Table 1. Details of each simulation in this research

Simulation Number	Number of Ensembles	Number of Students	Total Number of Networks
1	3	3	9
2	3	5	15
3	3	10	30
4	5	3	15
5	5	5	25
6	5	10	50
7	10	3	30
8	10	5	50
9	10	10	100
Total	54	54	324

5 Simulation Results

Nine simulations are run generating three sets of the 3-ensemble model, three sets of the 5-ensemble model, and three sets of the 10-ensemble model (see Table 1). Each student in all nine simulations is trained independently on *diabetes1* using CV1 and CV2. Performance evaluation focuses only on test data misclassification rates, and rate of disparity or improvement amongst models.

5.1. Simulation Results for the 3-Ensemble Model

According to the 3-ensemble model results presented in Table 2, while individual networks from Simulation 1, 2 and 3 misclassify the test data on average 31.57% of the time, the 3-ensemble offers an average misclassification rate of 27.72%. The 3-ensemble reflects an average misclassification reduction of 3.85, or 12.25%

improvement from its average individual student. The ensemble set exhibiting the best performance in this model is the 3-ensemble of ten students, with an average misclassification rate of 26.56%. This rate reflects a 15.09% over the average of its individual networks.

Table 2. The 3-ensemble model result averages

Simulation #	Average misclassification rate of individual students	Misclassification rate of ensemble	Ensemble rate improvement compared to the average misclassification rate of its individual students	
			Improved	% Improved
1- 3ens/3std	31.36	27.08	4.28	13.65
2- 3ens/5std	32.08	29.52	2.57	8.00
3- 3ens/10std	31.27	26.56	4.71	15.09
Average	31.57	27.72	3.85	12.25

5.2. Simulation Results for the 5-Ensemble Model

According to the 5-ensemble model results presented in Table 3, while individual networks from Simulation 4, 5 and 6 misclassify the test data on average 30.63% of the time, the 5-ensemble offers an average misclassification rate of 26.38%. The 5-ensemble reflects an average misclassification reduction of 4.25, or 13.95% improvement from its average individual student. The ensemble set exhibiting the best performance in this model is the 5-ensemble of ten students, with an average misclassification rate of 26.09%. This rate reflects a 16.21% improvement over that of its average individual student. Table 3 depicts only the best, worst, and average ensembles for each model, where best and worst are defined by the model improvement factor.

Table 3. The 5-ensemble model results

Simulation#	Ens	Average misclass. rate of individual students	Misclass. rate of ensemble	Ensemble rate improvement compared to the average misclass. rate of individual students	
				Improved	%Improved
4- 5ens/3std	Best	26.38	20.87	5.51	20.89
	Worst	34.49	32.17	2.32	6.73
	Avg	30.26	26.26	4.00	13.53
5- 5ens/5std	Best	30.26	24.35	5.91	19.53
	Worst	31.13	30.43	.7	2.25
	Avg	3.47	26.78	3.69	12.10
6- 5ens/10std	Best	32.09	25.22	6.87	21.41
	Worst	31.39	28.70	2.69	8.57
	Avg	31.15	26.09	5.06	16.21
Average		30.63	26.38	4.25	13.95

5.3. Simulation Results for the 10-Ensemble Model

According to the 10-ensemble model results presented in Table 4, while individual networks from Simulation 7, 8 and 9 misclassify the test data on average 29.97% of the time, the 10-ensemble offers an average misclassification rate of 25.03%. The 10-ensemble reflects a misclassification reduction of 4.94, or 17.47% improvement from its average individual student. The ensemble set exhibiting the best performance in this model is the 10-ensemble of ten students, with an average misclassification rate of 24.38%. This rate reflects a 20.11% improvement over the average of its individual networks.

Table 4. The 10-ensemble model results

Simulation#	Ens	Average misclass. rate of individual students	Misclass. rate of ensemble	Ensemble rate improvement compared to the average misclass. rate of individual students	
				Improved	%Improved
7- 10ens/3std	Best	28.65	19.30	9.35	32.64
	Worst	26.90	28.07	-1.17	-4.35
	Avg	30.35	25.79	4.56	15.05
8- 10ens/5std	Best	22.46	12.28	10.18	45.33
	Worst	39.65	36.84	2.81	7.09
	Avg	29.72	24.91	4.81	17.24
9-10ens/10std	Best	22.63	12.28	10.35	45.74
	Worst	34.39	36.84	-2.45	-7.12
	Avg	29.84	24.38	5.46	20.11
Average		29.97	25.03	4.94	17.47

6 Discussion of Results

6.1. Results of CV1

From Tables 2,3, and 4, it can be concluded that all CV1 ensemble models have a significantly lower misclassification rate than the average of their independent counterparts. While individual networks misclassify the test data on average 30.72% of the time (calculation based on overall averages from tables 2,3,4), the CV1 models average 26.38%. This average represents a reduction of 4.35, or a 14.55% performance increase. The model exhibiting the best performance is the 10-ensemble model, with an average misclassification rate of 25.03%. This rate reflects a 17.47% improvement over the average of its individual networks.

These results confirms the proposition that CV1 may be used to generate ensembles that demonstrate a performance improvement over individual networks. Two ensembles in Simulation 7 and two ensembles in Simulation 9 show an individual student average which outperforms their respective ensembles by a small margin (see Table 4 for details). However, the average performance of the models in Simulation 7 and Simulation 9 are consistent with the CV1 proposition, reflecting a

significant ensemble improvement of 15.05% and 20.11%, respectively. The most probable explanation for the performance degradation on these few ensembles can be attributed to a biased test set, based on its relative location. However, by rotating the test set for each ensemble of a given model CV2 reduces the impact of such bias.

6.2. Results of CV2

By convention, the unseen test data typically come from the bottom of the data set. Potential *bias* of a particular subset of the data, may inadvertently sabotage the testing process leading to inaccuracies in performance evaluation. A better measure of performance would test the model on as many different subsets of unseen data patterns as possible. Table 3 and Table 4 nicely illustrate the *bias* problem. In Table 4 Simulation 9, the ensemble performance varies by as much as 12.8 points, or 52.86%. The significance of these results provide support for the inclusion of cross-validation techniques in the model, for performance evaluation. The *bias* problem is clearly illustrated by the results presented for the 10-ensemble models with five or ten students, where the classification range for ensemble performance is as great as 24.56 percentage points. These two models have the best misclassification rate of 12.28% and the worst misclassification rate of 36.84%. Clearly, if the performance of a model were measured by a single test set it is likely not to be an accurate reflection of model performance [15][16]. These results and observations provide support for the significance of CV2 as both a valuable evaluation method and technique for experimentation efficiency in ensemble generation, testing, and analysis.

6.3. Secondary Findings

A secondary finding relates the misclassification rate of a given model to the number of ensembles in that model. Results show a steady decrease in misclassification rates when the number of ensembles in a given model is increased. This trend may be explained by the increase in available training data, based on the train-test split [4]. When the data set subdivisions increase in numbers they cause a decrease in the number of patterns per group. Further investigation is needed to provide conclusive evidence on the impact of the train-test split in CV1 and CV2 modeling. This investigation is left for future work.

7 Conclusions

The scope of this study represents only a small segment of the issues associated with ensemble model generation and evaluation. Limitations imposed on the experimental conditions provide avenues for future work. Deployment of future NNES versions will provide support for testing of multi-classification problems. Classification flexibility will encourage the continued evaluation of CV1 and CV2 within the NNES framework, as applied to other interesting domains. Another objective will seek expansion of the framework to include increased options for learning and combination methodologies. Future work will also explore the relationship between the number of ensembles in a given model and its relative performance.

In conclusion, although there is still much work to be done, significant progress has been made into the investigation of *two-level cross-validation* techniques for ensemble generation and evaluation. Ensembles generated using *level-one cross-validation* (CV1) are shown to provide a lower misclassification rate than their individual networks. Results support CV1 as a sound methodology for ensemble model generation. Likewise, simulation models using *level-two cross-validation* (CV2) provide a sound methodology for effectively evaluating the true performance of a model. Furthermore, ensemble investigations requiring large-scale experimentation have been simplified by this work and the deployment of CV1 and CV2 in NNES 4.0.

References

1. Amari, S., Murata, N., Muller, K.R., Finke, M., Yang, H.H.: "Asymptotic statistical theory of overtraining and cross-validation", *IEEE Transactions on Neural Networks*, Vol.8, No.15 (1997) 985-996
2. Fahlman, S.E., Lebiere, C.: "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Los Altos, CA: Morgan Kaufmann (1990) 524-532
3. Hansen, L.K., Salamon, P.: "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12 (1990) 993-1001
4. Kearns, M.: "A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split," *Neural Computation*, Vol. 9 (1997) 1143-1162
5. Krogh, A., Sollich, P.: "Statistical mechanics of ensemble learning," *Physical Review*, vol. E 55 (1997) 811
6. Krogh, A., Vedelsby, J.: "Neural network ensembles, cross-validation, and active learning," in *Advances in Neural Information Processing System 7*, G. Tesauro, D. S. Touretzky, and T.K. Leen, Ed. Cambridge, MA: MIT Press (1995) 231-238
7. Leisch, F., Jain, L.C., Hornik, J.: "Cross-validation with active pattern selection for neural-network classifiers," *IEEE Transactions on Neural Networks*, Vol.9, No.1 (1998) 35-41
8. Nowla, S., Rivals, I., Personnaz, L.: "On cross validation for model selection," *Neural Computation*, Vol. 11 (1999) 863-871
9. Prechelt, L.: "PROBEN1--A set of neural network benchmark problems and benchmarking rules," Universitat Karlsruhe, Karlsruhe, Germany, Technical Report (1994) 21-94
10. Renner, R.S.: *Improving Generalization of Constructive Neural Networks Using Ensembles*, Ph.D. dissertation, The Florida State University (1999)
11. Renner, R.S.: “Systems of ensemble networks demonstrate superiority over individual cascor nets”, in proceedings of the *International Conference on Artificial Intelligence* (2000) 367-373
12. Renner, R.S., Lacher, R.C.: “Combining Constructive Neural Networks for Ensemble Classification,” in proceedings of the *Joint Conference on Intelligent Systems* (2000) 887-891
13. Renner, R.S., Lacher, R.C., Juliano, B.J.: “A Simulation Tool for Managing Intelligent Ensembles”, in proceedings of the *International Conference on Artificial Intelligence* (1999) 578-584
14. Ripley, B.D.: *Pattern Recognition and Neural Networks*, Cambridge, MA: Cambridge University Press (1996)
15. Stone, M.: "Asymptotic for and against cross-validation," *Biometrika*, Vol. 64 (1977) 29-35
16. Stone, M.: "Cross-validation choice and assessment of statistical predictions," in *Journal of the Royal Statistical Society*, Vol. 36, No. 1 (1994) 111-147