Ensembles of General Regression Neural Networks for Short-Term Electricity Demand Forecasting

Grzegorz Dudek
Department of Electrical Engineering
Czestochowa University of Technology
42-200 Czestochowa, Al. Armii Krajowej 17, Poland
dudek@el.pcz.czest.pl

***Abstract*—This work presents ensembles of general regression neural network for short-term electricity demand forecasting. Several types of ensembles are proposed which differ in the source of diversity of individual members. Diversity is generated using different subsets of training data, different subsets of features, randomly disrupted training data and randomly disrupted model parameters. Experimental study on several datasets demonstrates that ensemble learning leads to decreasing in forecast errors comparing to the mean errors of the base learners.**

Keywords—ensemble forecasting; general regression neural network; pattern-based forecasting; short-term load forecasting

I. INTRODUCTION

Neural network (NN) ensemble is a learning paradigm where a set of NNs is learned for the same task. It originates from the pioneering work of Hansen and Salamon [1], where the authors have shown that the generalization error can be significantly reduced by invoking ensembles of similar networks. Ensemble learning systems in the past few years become a very hot topic in machine learning, computational intelligence and data mining communities. They have been successfully applied in many areas including time series forecasting.

Ensemble learning systems are composed of many base learners (NNs in our case), where each learner provides an estimate of a target function. These estimates are combined in some fashion to produce a common response, hopefully reducing the generalization error compared to a single learner. In regression ensembles used in time series forecasting the multiple estimates are integrated by a linear combination. The weights assigned to members in the linear combination can be the same for each member or can be dependent on the individual learner performance.

Recently there have been numerous proposals for creating ensemble of predictors. Two of the most popular are bagging [2] and boosting [3]. Bagging generates many training subsets from the original set and using these subsets trains component NNs. The outputs of the members are integrated to get ensemble output. Boosting generates a series of NNs whose training sets are determined by the performance of former ones. Training points which are predicted with higher errors by former NNs will play more important roles in the training of later NNs. Many other approaches for training the members of the ensemble appear in the literature in recent years.

The fundamental issue in ensemble learning is ensuring the diversity of learners. A good tradeoff between performance and diversity underlies the success of ensemble learning. The diversity can be exactly formulated in terms of the covariance between individual learner outputs, and the optimum level is expressed in terms of a bias-variance-covariance trade-off [4]. As it was shown in [5] the error of the ensemble will never be higher than the average error of the individual learners. Generating diverse learners is a challenging problem. Diversity can be achieved through several strategies. One of the most popular is learning on different subsets of the training set. Different sampling strategies lead to different ensemble algorithms. Using different subsets of the features to train each learner leads to random subspace methods [6]. Other common approaches also include using different parameters of the learners, such as number of hidden neurons or even using different base learners as the ensemble members (heterogeneous ensemble [7]). Some experimental results show that heterogeneous ensembles can improve accuracy compared to homogenous ones [8]. This is because the error terms of models of different types are less correlated than the errors of models of the same type.

In this work a homogeneous ensemble is proposed for short-term electricity demand forecasting. Ensemble members are GRNNs which are learned using different strategies to cause diversity. They include: different subsets of training data, different subsets of features, randomly disrupted training data and randomly disrupted model parameters.

The rest of this paper is organized as follows. The GRNN architecture and data representation in the forecasting model in Section II are described. Types of ensembles using different ways of diversity generation in Section III are proposed. Section IV presents results of the experimental study on four real-world data sets. Finally, some concluding remarks are drawn in Section V.

II. GRNN FOR STLF

A GRNN proposed by Specht [9] is a special case of Radial Basis Function neural network (RBFNN). The characteristic feature of RBFNN is that its activation functions

978-1-5090-6406-9/17/$31.00

are radially symmetric about the center vector. The neuron output nonlinearly decreases with the distance between an input vector and a centre vector. The activation function is commonly taken to be Gaussian:

$$G(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{s^2}\right) \quad (1)$$

where: $\mathbf{c}$ is a center vector, s is a RBF spread (or bandwidth) and ||.|| is a Euclidean norm.

In RBFNN case center vectors represent groups of neighboring training points $\mathbf{x}$ and are usually determined using some clustering algorithm. In GRNN case each center vector represents individual training point. So, GRNN is a memory-based network, where each learning point $\mathbf{x}_i$ is represented by one RBF neuron having activation function $G_i(\mathbf{x})$ with the center $\mathbf{c}_i = \mathbf{x}_i \in \Omega$, where Ω is a training set. Advantages of GRNN are: one pass, fast learning, easy tuning and highly parallel structure. The algorithm provides smooth approximation of a target function even with sparse data in a multidimensional space.

As it can be seen from Fig. 1 GRNN consists of four layers of nodes with entirely different roles:

- input layer, which distribute inputs $\mathbf{x}$ without processing,
- pattern (RBF) layer, where a nonlinear transformation is applied on the input to the hidden space (usually the hidden space is of higher dimensionality than the input space),
- summation layer, where two sums are calculated: 1) sum of target patterns $\mathbf{y}_i$ weighted by the neuron outputs, and 2) sum of the neuron outputs,
- output layer, expressing the weighted sum of target patterns.

A neuron output expresses similarity between the input vector $\mathbf{x}$ and the i-th training vector. So, the pattern layer maps n-dimensional input space into N-dimensional space of similarity. The higher similarity level, the higher i-th neuron output and consequently the higher contribution of the target pattern $\mathbf{y}_i$ to the prediction. The GRNN output is an average of target y-patterns weighted by the degree of similarity between paired with them training x-patterns and the input pattern $\mathbf{x}$:

$$\hat{\mathbf{y}} = g(\mathbf{x}) = \frac{\sum_{j=1}^{N} G_i(\mathbf{x})\mathbf{y}_i}{\sum_{i=1}^{N} G_i(\mathbf{x})} \quad (2)$$

A spread s is the only parameter to estimate. It determines the smoothness of the fitted surface and generalization performance of the model. As spread becomes larger the neuron output increases (y-pattern weight in (2) increases), with the result that the fitted function becomes smoother. A spread can be the same for all neurons or different for each neuron. Determining optimal spread values is a major problem in GRNN learning. In [10] for adjusting spread (the same for all neurons) simple enumerative method was used. In [11] to search N-dimensional spread space a differential evolution algorithm was applied.

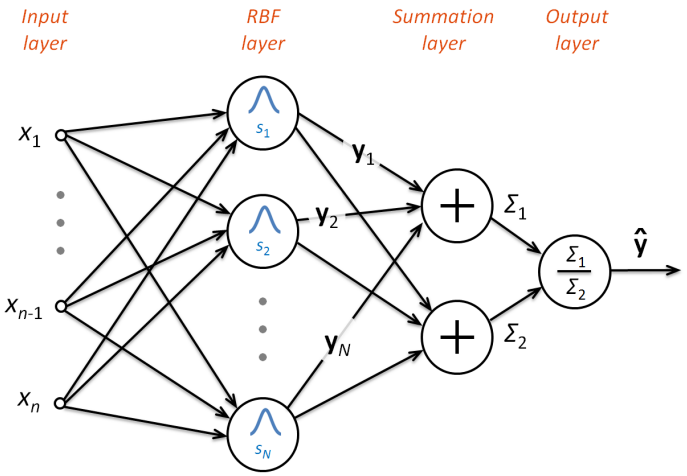

Fig. 1. GRNN architecture.

It is worth noting that GRNN generates a vector as an output. The dimension of this vector does not affect the number of parameters to estimate unlike in other popular models such as multilayer perceptron or neuro-fuzzy networks. This should be considered as a valuable property.

GRNN forecasting model works on patterns of the daily cycles of the load time series [12]. An input pattern $\mathbf{x}_i = [x_{i,1}\ x_{i,2}\ \ldots\ x_{i,n}]$ represents a daily period of the load time series: $\mathbf{L}_i = [L_{i,1}\ L_{i,2}\ \ldots\ L_{i,n}]$, where i is the day number and n is 24 for hourly resolution, 48 for half-hourly resolution or 96 for quarter-hourly resolution. An x-pattern is the normalized load vector $\mathbf{L}_i$ with components defined as follows:

$$x_{i,t} = \frac{L_{i,t} - \overline{L}_i}{D_i} \quad (3)$$

where $L_{i,t}$ is the power system load in period t of the day i, $\overline{L}_i$ is the mean load of the day i and $D_i = \sqrt{\sum_{l=1}^{n}(L_{i,l} - \overline{L}_i)^2}$ is the dispersion of the time series elements in the daily period i.

An output pattern $\mathbf{y}_i = [y_{i,1}\ y_{i,2}\ \ldots\ y_{i,n}]$ represents a forecasted daily load period for the day $i+\tau$: $\mathbf{L}_{i+\tau} = [L_{i+\tau,1}\ L_{i+\tau,2}\ \ldots\ L_{i+\tau,n}]$, where $\tau > 0$ is a forecast horizon. Its components are defined as follows:

$$y_{i,t} = \frac{L_{i+\tau,t} - \overline{L}_i}{D_i} \quad (4)$$

To forecast the daily curve for the day of the week δ, GRNN learns on the training set composed of input-output pattern pairs from history: $\Omega = \{(\mathbf{x}_i, \mathbf{y}_i) : type(\mathbf{y}_i) = \delta\}$, where $type(\mathbf{y}_i)$ is the day of the week of pattern $\mathbf{y}_i$. So, for the day-type δ GRNN learns on the training patterns representing only this type of the day. The forecasted load vector $\mathbf{L}_{i+\tau}$ is

calculated from transformed equation (4) after the forecast of pattern **y** is generated by the model (decoding phase):

$$\hat{L}_{i+\tau,t} = \hat{y}_{i,t} D_i + \overline{L} \tag{5}$$

Due to coding daily periods of the load time series as x- and y-patterns we unify the input and output data and simplify relationships between them. Note that x- and y-patterns express unified shapes of daily curves. Annual and weekly cycles and also trend are filtered out. This is further discussed in [12]. The expected result of using patterns is a simpler and more accurate forecasting model.

III. Ensembles of GRNN

We are given a set of M individual learners $\{h^1, h^2, ..., h^M\}$. The forecasting tasks that each learner solves are: on the basis of input data predict $\mathbf{L}_{i+\tau}$, i.e. the load daily curve for the day $i+\tau$. Let us denote the output vector of h^k for the specific forecasting task as $\hat{\mathbf{y}}_i^k \in \mathrm{R}^n$. Having the predicted y-pattern, from (5) we calculate the forecast of the daily curve for the day $i+\tau$ generated by the k-th member: $\hat{\mathbf{L}}_{i+\tau}^k$. We combine the vectors $\hat{\mathbf{L}}_{i+\tau}^k$ to attain the final prediction using simple averaging. The combined ensemble output is of the form:

$$\hat{\mathbf{L}}_{i+\tau} = \frac{1}{M} \sum_{k=1}^{M} \hat{\mathbf{L}}_{i+\tau}^k \tag{6}$$

As previously mentioned in Section I a key issue in ensemble learning is to ensure diversity of individual learners. It was achieved in the proposed GRNN ensemble using the following strategies:

D1. Learning on different subsets of the training data. For each ensemble member a random sample without replacement of size $N' < N$ from the training set Ω is selected. The spread parameters for each neuron of each member are the same:

$$s = a \cdot d_{5NN} \tag{7}$$

where: a = const > 0, d_{5NN} is the mean distance between each $\mathbf{x} \in \Omega$ and its five nearest neighbors in Ω.

D2. Learning on different subsets of features (random subspace method [6]). For each ensemble member the features are randomly sampled without replacement. The sample size is $n' < n$. The spread is determined as:

$$s = a \cdot d_{5NN} \sqrt{\frac{n'}{n}} \tag{8}$$

The factor $(n'/n)^{0.5}$ corresponds to the reduction in Euclidean distance between x-patterns in n'-dimensional space relative to n-dimensional space.

D3. Learning using spread parameters randomly disrupted. The initial value of spread, which is the same for all neurons of all members, is randomly perturbed by Gaussian noise:

$$s_{k,i} = a \cdot d_{5NN} \cdot \xi_{k,i} \tag{9}$$

where $\xi_{k,i}$ are random numbers drawn from normal distribution N(1, σ_s) for neuron i of the member k.

The level of noise is controlled by the standard deviation σ_s.

D4. Learning using x-patterns randomly disrupted. Each training pattern $\mathbf{x} \in \Omega$ is perturbed by Gaussian noise:

$$x_{i,t} = x_{i,t} \cdot \xi_{i,t} \tag{10}$$

where $\xi_{i,t} \sim$ N(1, σ_x).

Standard deviation σ_x controls the noise level. The spread parameters are determined using (7).

D5. Learning using y-patterns randomly disrupted. Each training pattern $\mathbf{y} \in \Omega$ is perturbed by Gaussian noise:

$$y_{i,t} = y_{i,t} \cdot \xi_{i,t} \tag{11}$$

where $\xi_{i,t} \sim$ N(1, σ_y).

Standard deviation σ_y controls the noise level. The spread parameters are determined using (7).

IV. Simulation Study

The proposed approach using GRNN ensembles for STLF was examined on four load time series:

- PL: time series of the hourly load of the Polish power system over the period 2002–2004. The test set includes data from 2004 with the exception of 13 atypical days (e.g. public holidays),
- FR: time series of the half-hourly load of the French power system over the period 2007–2009. The test set includes data from 2009 except for 21 atypical days,
- GB: time series of the half-hourly load of the British power system over the period 2007–2009. The test set includes data from 2009 except for 18 atypical days,
- VC: time series of the half-hourly load of the power system of Victoria, Australia, over the period 2006–2008. The test set includes data from 2008 except for 12 atypical days.

In all cases the training sets contained data from the first two years. The problem is to forecast the system hourly load (for PL) or half-hourly load (for FR, GB and VC) for the next day (τ = 1). The forecasts were generated independently by each of M ensemble members. Then the forecasts were combined using (6). Five methods of diversity generation described in Section III were applied. The following parameters of the ensemble members were used:

- M = 100,
- a = 0.6,
- N' = 2/3N,
- n' = 2/3n,

- $\sigma_s = \sigma_x = \sigma_y = 0.15$.

Examples of forecasts of each member and the final forecast of ensemble for the five methods of diversity generation D1-D5 in Fig. 2 are shown. As we can see from these figures a collection of M forecasts shows different diversity. The diversity is measured using standard deviation of the forecasts of individual members. The standard deviation of the forecasts for the day $i+\tau$ is defined as:

$$std_{i+\tau} = \frac{1}{n}\sum_{t=1}^{n}\sqrt{\frac{1}{M-1}\sum_{k=1}^{M}(\hat{L}^{k}_{i+\tau,t} - \hat{L}_{i+\tau,t})^2} \qquad (12)$$

where $\hat{L}^{k}_{i+\tau,t}$ is the t-th component of the load vector forecasted by the k-th member and $\hat{L}_{i+\tau,t}$ is the t-th component of the mean load vector forecasted by the ensemble (6).

Results of forecasting in Table I are shown, where:

- $MAPE_{ens}$ is an error of ensemble for the test set,
- $MAPE_{mem}$ is an mean error of the members for the test set,
- $\overline{std}$ is a mean standard deviation (12) for all forecasted tasks from the test set.

Depending on the source of diversity different level of diversity is observed. Obviously, it is dependent also on the parameters of the method of diversity generation, such as: random sample size in D1, number of features sampled in D2, and the standard deviation of the Gaussian noise disturbing data or spreads in D3-D5. Higher values of these parameters cause more diverse outputs of ensemble members. But too high values lead to deterioration in performance of the ensemble.

It can be seen from Tables I-IV that the errors for different sources of diversity are similar. It is hard to indicate the best strategy for the member diversification.

Fig. 3 shows distributions of errors generated by members and errors of ensembles for different methods of diversity generation. Note that error of ensemble is always lower than mean error of individual members. In most cases it is even lower than the error of the best member. The highest errors of members for D5 are observed. In this scenario the members are the least diverse as well. But the final ensemble forecasts for D5 do not differ in terms of errors from forecasts generated by other ensembles.

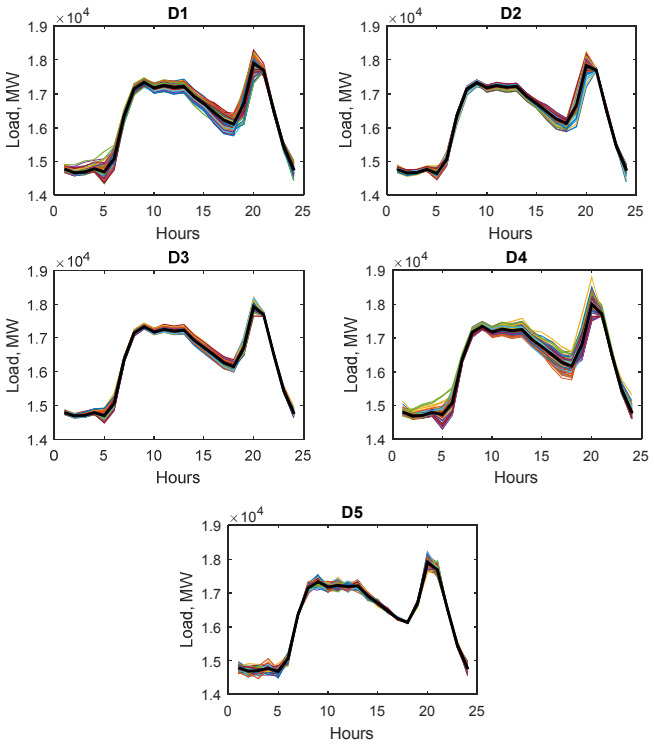

Fig. 2. Examples of forecasts generated by each member and the final forecast of ensemble (thick black line) for the five methods of diversity generation.

TABLE I. RESULTS FOR PL DATA

	D1	D2	D2	D4	D5
$MAPE_{ens}$	1.39	1.41	1.39	1.37	1.39
$MAPE_{mem}$	1.48	1.48	1.43	1.47	1.52
$\overline{std}$	98.90	77.54	61.24	97.04	106.19

TABLE II. RESULTS FOR FR DATA

	D1	D2	D2	D4	D5
$MAPE_{ens}$	1.63	1.67	1.65	1.65	1.66
$MAPE_{mem}$	1.76	1.70	1.69	1.76	1.81
$\overline{std}$	411.32	196.65	236.25	387.16	401.45

TABLE III. RESULTS FOR GB DATA

	D1	D2	D2	D4	D5
$MAPE_{ens}$	1.60	1.64	1.60	1.59	1.62
$MAPE_{mem}$	1.70	1.67	1.64	1.89	2.01
$\overline{std}$	232.18	114.96	139.61	450.78	418.75

TABLE IV. RESULTS FOR VC DATA

	D1	D2	D2	D4	D5
$MAPE_{ens}$	2.78	2.83	2.79	2.79	2.80
$MAPE_{mem}$	2.93	2.88	2.87	2.88	2.92
$\overline{std}$	65.57	32.12	45.55	52.57	46.21

V. Conclusion

In this work homogeneous ensembles composed of GRNNs for short-term electricity demand forecasting are proposed. Five methods of diversity generation of ensemble members are analyzed. In experimental study the proposed ensembles have been evaluated on four real-world data sets. The results showed that for all considered sources of diversity the ensemble errors were lower than the mean errors of individual members, and in most cases even lower than error of the best member.

GRNNs as an ensemble members in the proposed approach have fixed parameters (spreads) which are not learned. The construction of the ensemble in such a case is much faster because it does not require tuning parameters of the base models. The ensemble parameters deciding about the diversity of the members such as: size of the random learning samples, number of features or parameters of the noise disturbing data or spreads, were also fixed in the proposed approach. But they can be optimized which should lead to further improvement in the performance.

References

[1] L.K. Hansen, P. Salamon, "Neural network ensembles," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993-1001, 1990.

[2] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123-140, 1996.

[3] Y. Freund, "Boosting a weak algorithm by majority," Information and Computation, vol. 121, no. 2, pp. 256- 285, 1995.

[4] G. Brown, J.L. Wyatt, P. Tino, "Managing diversity in regression ensembles" Journal of Machine Learning Research, vol. 6, pp. 1621–1650, 2005.

[5] A. Krogh, J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, Advances in Neural Information Processing Systems 7, pp. 231–238. MIT Press, Cambridge, MA, 1995.

[6] T.K. Ho, "The random subspace method for constructing decision forests," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832-844, 1998.

[7] G. Dudek, "Heterogeneous ensembles for short-term electricity demand forecasting," Proc. 17th Conf. Electric Power Engineering (EPE'2016), pp. 1-6, 2016.

[8] J. Wichard, C. Merkwirth, and M. Ogorzałek, "Building ensembles with heterogeneous models," in Course of the International School on Neural Nets, 2003.

[9] D.F. Specht, "A general regression neural network," IEEE Transactions on Neural Networks, vol. 2, no. 6, pp. 568–576, 1991.

[10] G. Dudek, "Neural networks for pattern-based short-term load forecasting: A comparative study," Neurocomputing, vol. 2015, pp. 64-74, 2016.

[11] G. Dudek, "Generalized regression neural network for forecasting time series with multiple seasonal cycles," In Filev D. et al. (eds.): Intelligent Systems'2014, Advances in Intelligent Systems and Computing 323, pp. 839-846, 2015.

[12] G. Dudek, "Pattern similarity-based methods for short-term load forecasting – Part 1: Principles," Applied Soft Computing, vol. 37, pp. 277-287, 2015.

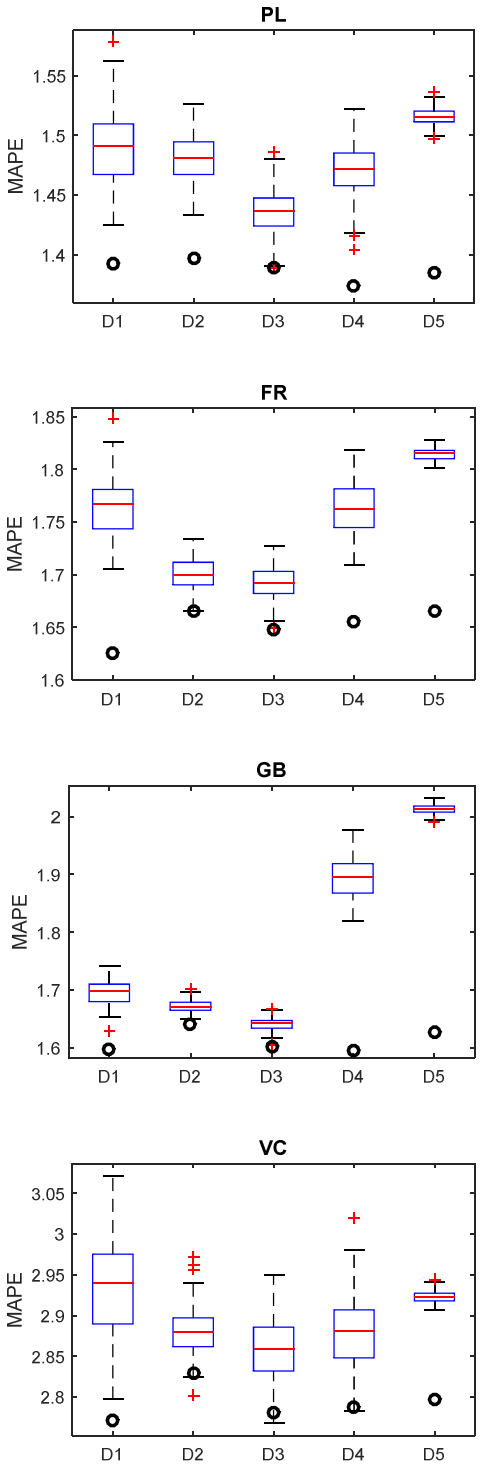

Fig. 3. Error distributions of individual members for five ways of diversity genaration; black rings show errors for ensembles.