

# An experiment on the evolution of an ensemble of neural networks for streamflow forecasting

M.-A. Boucher, J.-P. Laliberté, and F. Anctil

Chaire de recherche EDS en prévisions et actions hydrologiques, Département de génie civil, Université Laval, Pavillon Pouliot, Québec, G1K 7P4, Canada

Received: 9 September 2009 – Published in Hydrol. Earth Syst. Sci. Discuss.: 6 October 2009

Revised: 9 March 2010 – Accepted: 19 March 2010 – Published: 30 March 2010

**Abstract.** We present an experiment on fifty multilayer perceptrons trained for streamflow forecasting on three watersheds using bootstrapped input series. This type of neural network is common in hydrology and using multiple training repetitions (ensembling) is a popular practice: the information issued by the ensemble is then aggregated and considered to be the final output. Some authors proposed that the ensemble could serve the calculation of confidence intervals around the ensemble mean. In the following, we are interested in the reliability of confidence intervals obtained in such fashion and in tracking the evolution of the ensemble of neural networks during the training process. For each iteration of this process, the mean of the ensemble is computed along with various confidence intervals. The performance of the ensemble mean is evaluated based on the mean absolute error. Since the ensemble of neural networks resemble an ensemble streamflow forecast, we also use ensemble-specific quality assessment tools such as the Continuous Ranked Probability Score to quantify the forecasting performance of the ensemble formed by the neural networks repetitions. We show that while the performance of the single predictor formed by the ensemble mean improves throughout the training process, the reliability of the associated confidence intervals starts to decrease shortly after the initiation of this process. While there is no moment during the training where the reliability of the confidence intervals is perfect, we show that it is best after approximately 5 to 10 iterations, depending on the basin. We also show that the Continuous Ranked Probability Score and the logarithmic score do not evolve in the same fashion during the training, due to a particularity of the logarithmic score.

## 1 Introduction

Neural networks are used in hydrology since the 1990's (e.g. Kang et al., 1993; Karunanithi et al., 1994; Campolo et al., 1999; Tokar and Johnson, 1999). They have also been the object of some experiments in meteorology (e.g. Hsieh and Tang, 1998; Valverde Ramírez et al., 2005) and climatology (e.g. Knutti et al., 2003). Although it can be argued that neural networks models cannot contribute to the understanding of the processes at hand and that they are most often over parametrized, they remain very useful as simple, rapidly implemented, rainfall-runoff models.

One of the most frequently used neural network architecture in water resources research (e.g. Coulibaly et al., 1999; Maier and Dandy, 2000; Singh and Deo, 2007) is the multilayer perceptron (Rosenblatt, 1958). It is capable of learning any multivariate non-linear relationship between input and output values, if provided with a database of sufficient length and if satisfactory training is performed (Cybenko, 1989; Hornik et al., 1989). However, it is rarely the case that only one network is created and trained to solve a specific problem (e.g. Iyer and Rhinehart, 1999). Since the 90's (e.g. Hansen and Salamon, 1990) it has been proposed to train an ensemble of neural networks for each problem at hand. Subsequently, Breiman's bagging (Breiman, 1996), Shapire and Freund boosting (e.g. Freund and Shapire, 1996; Shrestha and Solomatine, 2006a) and other similar or derived techniques enforced the ensembling practices among the machine learning community. Ensembling is one of the available strategies to improve generalization capacity. This is based on the assumption that the gradient descent optimization for a single neural network can fall into a local minimum and therefore not provide the best solution. Training multiple networks from various random starting points provides a better coverage of the parameter space. The individual neural models forming the ensemble are usually aggregated to



Correspondence to: M.-A. Boucher  
(marie-a.boucher.1@ulaval.ca)

**Table 1.** Characteristics of the watersheds and corresponding databases used in the experiment.

Basin	Area km <sup>2</sup>	Daily precipitation (mm)		Daily streamflow (mm)		Database length (days)	
		Mean	St. dev.	Mean	St. dev.	Training	Validation
La Golo	930	3.99	6.84	2.40	2.42	2176	2131
Serein	1120	2.31	4.12	0.61	0.86	5225	5231
Leaf	1949	3.92	10.14	1.37	2.90	4895	4917

provide a single final output, either by simple or weighted averaging, by a regression between ensemble members or by other more sophisticated means (e.g. Freitas and Rodrigues, 2006).

Some authors also suggested that an ensemble could be used to issue confidence intervals to be associated with the forecast (e.g. Lajbcygier and Connor, 1997; Papadopoulos and Edwards, 2001; Shrestha and Solomatine, 2006b). However, it is our opinion that the reliability of such confidence intervals has rarely been investigated.

The recent development of bayesian neural networks (e.g. Mackay, 1992; Neal, 1996) and their successful application in probabilistic hydrological forecasting (e.g. Khan and Coulibaly, 2006; Kingston et al., 2005) suggest that they are the most appropriate tools to achieve reliable ensemble and probabilistic hydrological forecasting with neural networks. Nevertheless, MLPs remain very popular among hydrologists for their simplicity of implementation and because they produce very accurate forecasts for a wide range of situations. Therefore, it can be interesting to have a closer look at an ensemble of MLPs, as it evolves during training, and to assess the reliability of the probabilistic distribution they collectively form. Besides bayesian networks, there exist some other experiments regarding probabilistic-type forecasts with neural networks, such as the work by Carney et al. (2005), in which they use ensembles of Mixture Density networks to issue probabilistic surf height forecasts.

Here we propose an experiment where we follow an ensemble of MLPs during their training process, in a one-day-ahead streamflow forecasting situation. As the MLP ensemble evolves, we will investigate their probabilistic performance and compare it to the deterministic performance of a single predictor formed by averaging the individual neural network outputs. We will also pay close attention to the reliability of the confidence intervals computed from the ensembles. Because MLP ensembles resemble ensembles issued by an operational hydrological forecasting system, we will resort to typical ensemble-based performance assessment tools.

The remaining of the paper is divided as follows: the context of application is described in the next section, with a short description of the watersheds and corresponding databases. The subsequent section presents the protocol of

experiment, including the neural networks architecture, the ensemble construction methodology and the criteria used for performance evaluation. Results are presented in Sect. 4 along with a discussion on the relevant findings of this work. The paper ends with concluding remarks and perspectives at Sect. 5.

## 2 Context of application

### 2.1 The Leaf, Serein and Le Golo watersheds

The investigation described in this paper relies on databases for three watersheds with a residence time of the order of three days, representing different hydrological behaviours. A summary of the information related to them is provided in Table 1, while hydrographs are drawn in Fig. 1.

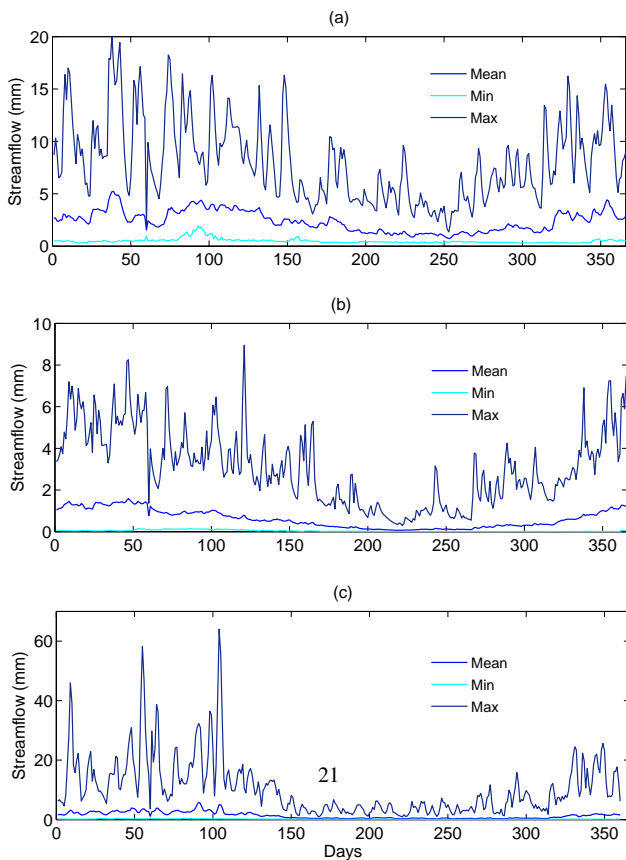
The Le Golo River is located in Corsica, France. There are many gauging stations along this river, which is the greatest of the island. The one with the longest record, located in Volpajola near the basin's outlet, will be used. This mountainous basin generates on some occasions relatively high streamflow in summer, while flow is usually maximal during winter and spring (Fig. 1a). The Serein River (Fig. 1b) is an unregulated tributary of the Yonne River, which joins the Seine River upstream of Paris. It exhibits a strong seasonal cycle (see Fig. 1b). Finally, the Leaf River is located near Collins, Mississippi, USA. Although this watershed also exhibits a seasonal cycle (see Fig. 1c), it is not very pronounced.

All data are standardized before being fed to the neural networks. This procedure ensures that all input data have the same range of values.

## 3 Protocol of experiment

### 3.1 Separation of the databases in subsets

The database for each basin was divided in a training and a testing datasets using a Kohonen network or self organizing map (Kohonen, 1990). It is a clustering method which employs a network formed of two layers (input and output). The input layer receives the data and the neurons of the output layer, structured to form a map, are the equivalent of



**Fig. 1.** Daily mean, maximum and minimum streamflows for the (a) Le Golo (b) Serein and (c) Leaf Rivers.

clusters. The observations are therefore distributed in those clusters according to their similarities. The number of output neurons (clusters) must be determined by a calibration process. Here we use the same Kohonen network as Ancil and Lauzon (2004). After testing for many configurations of the output map, they determined that a  $3 \times 3$  map was optimal. Once the nine clusters are identified, two subsets of identical size are created by randomly selecting daily events within each cluster. This ensures that the training dataset is statistically equivalent to the testing dataset, thus avoiding, for example, that the training set comprises many large streamflow events while the testing set contains few. A small experiment was also carried out where the database for each basin was split in two halves in order to maintain the chronological order in the training and rainingesting databases. Various statistics (mean, standard deviation, minimum and maximum value, kurtosis and skewness) were computed for those two datasets as well as for the training and testing datasets used in the experiment presented in this paper. Although the chronologically ordered datasets did not have enormous disparities in their statistics, the training and testing datasets obtained using the self organizing map were even more similar, with almost identical statistics.

The temporal correlation in the data is preserved even if the chronological order of the original series is not. All watersheds used in this study have a response time of about three days. Because multilayer perceptrons (see Sect. 3.2) do not account for the temporal correlation between the inputs and the output, it has to be recreated artificially. To achieve this, we first use the entire database, with all entries in chronological order, to produce a  $n$  by 5 matrix,  $n$  being the number of streamflow observations in the whole database. The fifth column is the observed streamflow. The first three columns are the observed precipitation values for the three previous days. The fourth column is the observed streamflow for the previous day. Then, the Kohonen mapping, separation of the database and bootstrap are performed using the row indices, ensuring that the observed streamflow is accompanied by the appropriate previous data in chronological order. Because the response time is limited (three days), there is no need to provide the network with observations further in the past.

### 3.2 Basic neural network architecture

The MLPs used to conduct this study comprise three layers: the input layer, the hidden layer in which there are five neurons, and the output layer in which there is only one neuron. The input layer is constituted of the observed streamflow ( $Q$ ) at the present time  $t$  and the precipitation ( $P$ ) at times  $t$ ,  $t-1$  and  $t-2$ . The output neuron issues  $Q_{t+1}$ , the one-day-ahead streamflow. The number of hidden neurons and input selection follows the trial and error process performed in the work by Ancil and Lauzon (2004) on the same data sets. Each input vector is connected to each neuron in the hidden layer and each neuron in the hidden layer is connected to the output neuron. A weight value ( $W_{i,j}$  or  $W_{j,k}$ ), where  $i$ ,  $j$  and  $k$  represent, respectively, the input number, the neuron number and the output number is assigned to each of these links. A bias ( $b_j$ ) value is also assigned to each neuron of the hidden and output layers. The weights and biases are the parameters of the neural model. They are randomly initialized and then an iterative optimization process is performed until the outputs of the model match the recorded observed streamflow data. Each iteration is called an “epoch” and the optimization algorithm is Levenberg-Marquardt Backpropagation (Levenberg, 1944; Marquardt, 1963).

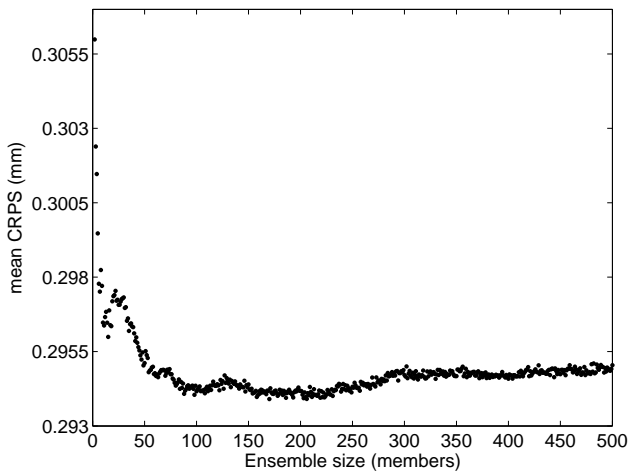
The transfer function for the neurons in the hidden layer is the sigmoid tangent, given by

$$C(\xi) = \frac{2}{1 + e^{-2\xi}} - 1, \quad (1)$$

where  $\xi$  represents the weighted sum of input vectors plus bias  $b_j$  and is given by

$$\xi_j = P_t W_{1,j} + P_{t-1} W_{2,j} + P_{t-2} W_{3,j} + Q_t W_{4,j} + b_j. \quad (2)$$

A linear transfer function is used for the output neuron in accordance with the Universal Approximation Theorem (Hornik et al., 1989).



**Fig. 2.** Variation of the CRPS with the number of neural networks in the ensemble (Le Golo River).

The architecture described above is very simple, but produces unbounded models. They are good interpolators, but have uncontrolled extrapolation capacities. When predicting streamflow, the absence of a lower limit for the forecast sometimes causes the network to issue negative values. Crespo et al. (1993) chose to replace those negative forecasts with zeros. In this study, we chose to replace them by the smallest archived streamflow observation for each watershed.

### 3.3 Ensemble construction

An ensemble formed of the outputs of 50 randomly initialized and individually trained MLPs was set up for every watershed: a strategy that is largely inspired from bagging (Breiman, 1996), in which each of the 50 forecasted time series is called a member of the ensemble. Therefore, the procedure asks for 50 bootstrapped training datasets (Efron and Tibshirani, 1993), a convenient way of accounting, at least to some extent, for the uncertainty component linked to the observations (e.g. Zhang et al., 2009; Ajami et al., 2007). The produced ensembles thus combine sources of uncertainty associated to the observations, through the bootstrap, and sources of uncertainty associated to the model, through a multi-model approach.

During training, the networks' parameters are stored for each epoch in order to be able to apply the partially trained networks to the testing dataset. The latter is not used at any moment of the training process, respecting Klemeš (1986) split sample strategy.

The choice of producing 50 members was determined experimentally. Using the databases at hand, the mean CRPS, which is explained later at Sect. 3.4, was computed for ensembles up to 500 members. Even if the mean CRPS does continue to decrease and stabilizes at about 200, we believe that fifty members are deemed sufficient to provide an

accurate estimation, as illustrated by Fig. 2. Coincidentally, the ECMWF operational meteorological ensemble prediction system consists of fifty members.

The bootstrap strategy has been confronted with two other options: producing only 5 or 10 bootstrapped series, which means that each new series are used to train 10 or 5 networks, respectively. Results of this test are not reported here because no clear distinction could be identified in terms of CRPS or other score values.

### 3.4 Multicriteria evaluation of performance

In order to assess the performance of the ensemble at each epoch of the learning process, we used the following ensemble-specific quality evaluation tools. First, we used numerical criteria, namely the Continuous Ranked Probability Score (CRPS) and its corresponding decomposition (Hersbach, 2000) as well as the logarithmic score (e.g. Good, 1952). We also used graphical quality assessment tools: the rank histogram (Talagrand et al., 1997; Hamill and Colucci, 1997) and reliability diagram (e.g. Wilks, 1995). Since these tools are described in great details in references such as Wilks (1995) and Jolliffe and Stephenson (2003), only a short description is provided hereafter.

We adopt the point of view of Gneiting and Raftery (2007), according to which a good ensemble forecast maximizes sharpness, subject to calibration. While sharpness refers to the precision of the ensemble members, calibration refers to the statistical consistency between the forecasts and the observations. An ensemble forecasting system is sharp if all the members of the ensemble are close to the observed value. This ensemble forecasting system is also well calibrated (reliable) if the dispersion of the ensemble reflects the true uncertainty of the situation.

In this view, reliability (calibration) precedes precision (sharpness) in ensemble or probabilistic forecasting since the final goal is most often expressing the forecast in terms of a probability or providing the user with a way of assessing the uncertainty on the next outcome. Therefore, however precise an ensemble forecast may be, if it is not reliable it is not useful.

#### 3.4.1 The continuous ranked probability score

Let  $F(x)$  be the cumulative distribution function (cdf) fitted from the ensemble members,  $x$  the predicted values at time  $t$ ,  $x_{\text{obs}}$  the observed value at the same time, and  $1$  the indicator function. The CRPS consists in an integral of the difference between two cumulative distributions. It is defined as

$$\overline{\text{CRPS}}(F, x_{\text{obs}}) = \frac{1}{N} \sum_{t=1}^N \int_{-\infty}^{\infty} (F_t(x) - 1_{\{x \geq x_{\text{obs},t}\}})^2 dx, \quad (3)$$

where  $N$  is the number of forecast-observation pairs. The CRPS has to be averaged over many realisations to make up

for the fact that it is a comparison between a distribution and a scalar.

Gneiting and Raftery (2007) have formally demonstrated that the CRPS for probabilistic forecasts is equivalent to the mean absolute error (MAE) for single forecasts. This result is based on previous mathematical proofs by Baringhaus and Franz (2004) and by Székely and Rizzo (2005). It thus provides a convenient way to compare the performance of ensemble forecasts (mean CRPS) with the performance of single forecasts (MAE) for the same watershed. Here, the MAE is calculated using the average of all members of the NN ensemble as a single forecast.

Like for the MAE, the lower the CRPS, the better it is. The lower bound is zero for both. However, the CRPS and MAE values are directly proportional to the absolute value of the observation.

An interesting characteristic of the CRPS is that it can be decomposed in two components (Hersbach, 2000).

$$\overline{\text{CRPS}} = \overline{\text{Rel}} + \overline{\text{CRPS}}_{\text{Pot}}, \quad (4)$$

where  $\overline{\text{Rel}}$  is the reliability component and  $\overline{\text{CRPS}}_{\text{Pot}}$  is the potential mean CRPS. The former quantifies the extent to which the spread of the ensemble is really representative of the uncertainty associated with the forecasting situation, while the latter is the mean CRPS value that would be attained if the system was made perfectly reliable. This second component depends mostly on the data and on the choice of the model used to issue the forecasts.

This decomposition is based on the empirical cdf of the ensemble of neural networks. The two components are calculated with Eqs. (5 and 6)

$$\overline{\text{Rel}} = \sum_{k=0}^n \bar{g}_k (\bar{o}_k - P_k)^2, \quad (5)$$

$$\overline{\text{CRPS}}_{\text{Pot}} = \sum_{k=0}^n \bar{g}_k \bar{o}_k (1 - \bar{o}_k), \quad (6)$$

where  $P_k = \frac{k}{n}$  is the empirical cdf. The subscript  $k = 0, 1, \dots, 50$  refers to the sorted ensemble members.  $\bar{g}_k$  and  $\bar{o}_k$  are calculated using

$$\bar{g}_k = \bar{\alpha}_k + \bar{\beta}_k \quad (7)$$

and

$$\bar{o}_k = \frac{\bar{\beta}_k}{\bar{\alpha}_k + \bar{\beta}_k}, \quad (8)$$

where  $\bar{\alpha}_k$  and  $\bar{\beta}_k$  represent, respectively, the mean difference (in mm) between two forecasts in the sorted ensembles, for streamflow values inferior or superior to the observation.

### 3.4.2 The logarithmic score

The logarithmic score, or ignorance score (e.g. Roulston and Smith, 2002), is the logarithm of the probability density,  $f(x_{\text{obs}})$ , associated with the observed value. Consequently, a gamma pdf was fitted to the ensemble of neural networks at every time step for all basins and all epochs using the maximum likelihood estimation. Let  $S$  be the score value,  $f(\mathbf{x})$  the predictive distribution and  $x_{\text{obs}}$  the observed value. Therefore, the value taken by the score is:

$$S(f(\mathbf{x}), x_{\text{obs}}) = -\log(f(x_{\text{obs}})) \quad (9)$$

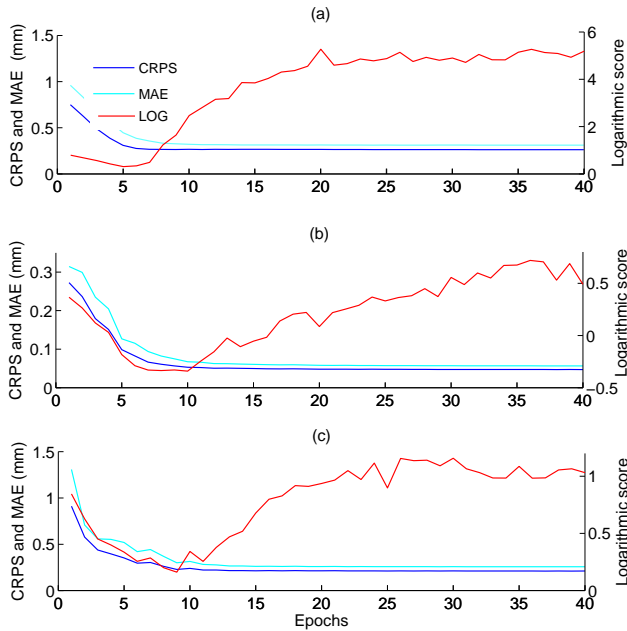
We use the logarithmic score in the negative orientation for reasons of coherence with the MAE and the CRPS. However, there is no lower bound for this score. In addition, when the observation falls outside of the predictive distribution, the corresponding probability density is zero. This produces an infinite value, which affects the calculation of the mean score. Here, we chose to replace those individual infinite scores by the next worst non-infinite value.

### 3.4.3 The rank histogram

The principle behind the rank histogram lies in the fact that if the ensemble forecasts are well calibrated, the observed value could be considered as a supplementary member of the ensemble. The construction of such a histogram is simple (Talagrand et al., 1997). The observed value at time  $t$  is first added to the corresponding forecasted ensemble and this new ensemble is sorted. For each forecast-observation pair, the rank of the observation is stored. Then, those ranks are plotted in a histogram. In the perfect case, this histogram is flat, so all ranks have equal relative frequency. A ‘‘U’’ shaped rank histogram indicates that the predictive distribution is underdispersed, so the observation falls outside the ensemble. Conversely, if the rank histogram has an arched form, it means that the distribution is overdispersed. If the rank histogram is asymmetric, the observation occupies some ranks more frequently than others. It can point out a bias in the forecasts.

### 3.4.4 The reliability diagram

The reliability diagram (e.g. Wilks, 1995), allows another visual assessment of the reliability of the forecasting system. For each ensemble, the limit values of confidence intervals are computed from 10 to 90% coverage by increment of 10%. The corresponding widths of these intervals are also computed. For each confidence interval and each forecast, it is verified whether or not the observed value is located inside the interval. Then, from the total number of occasions the observation is found to be inside each confidence interval, the effective coverage of the intervals are evaluated. The width of the confidence intervals are then plotted against the corresponding mean effective coverage. A reliable forecasting system would show a good correspondence between



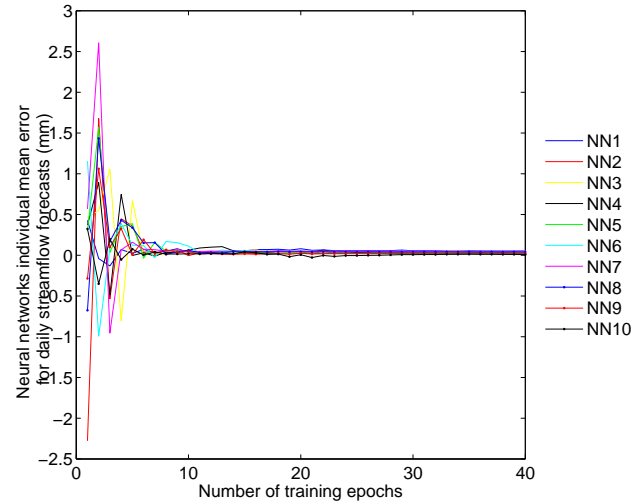
**Fig. 3.** CRPS, MAE and logarithmic score as a function of the number of training epochs for (a) Le Golo, (b) Serein, and (c) Leaf Rivers.

the nominal and effective coverage values, meaning that the nominal level of confidence of each interval corresponds to the effective coverage. In addition, for two equally reliable forecasting systems, this diagram allows the user to choose the system with the best resolution, that is, the one which provides the shortest widths for confidence intervals.

Over and under dispersion problems can also be diagnosed using the reliability diagram. Overdispersion corresponds to a situation where effective coverage is greater than nominal coverage of the confidence intervals. Underdispersion corresponds to opposite situation.

#### 4 Results

As explained in Sect. 3.4.1, the CRPS reduces to the absolute error in the case of a deterministic forecast, which allows the direct comparison of the performance of ensemble and deterministic forecasts for the same watershed (e.g. Velázquez et al., 2009). It is common practice to generate an ensemble of neural networks and to aggregate their outputs to form a deterministic forecast. Here, the performance of this deterministic forecast was compared with the performance of the ensemble (not averaged) forecast. Results are shown in Fig. 3, which presents the mean CRPS, MAE and mean logarithmic score calculated on the testing set. For all basins and every epoch, the CRPS values are lower than the MAE values. It indicates that the ensemble of neural networks performs better when taken as a whole than when aggregated in a single averaged predictor.

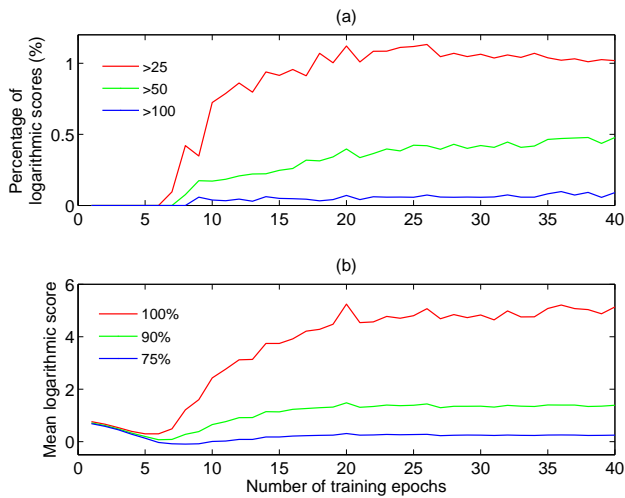


**Fig. 4.** Mean error as a function of the number of training epochs for ten neural networks (Le Golo River).

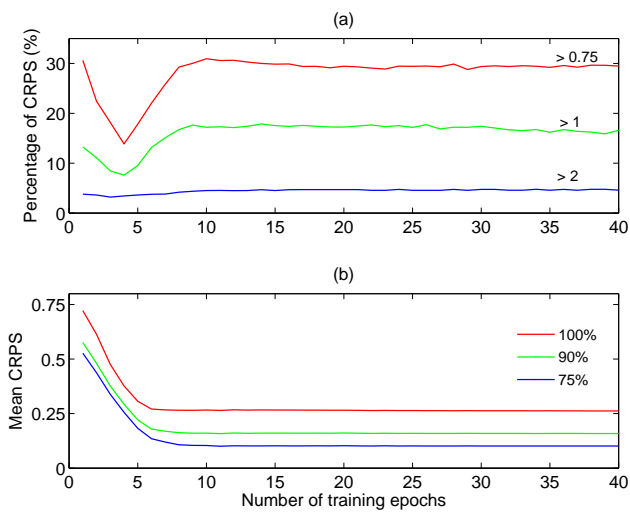
As expected, because of random initialization, the first training epochs offer poor performance (Fig. 3). However, this improves rapidly over the next training epochs, before reaching a plateau with a small negative slope. This indicates that at the beginning of the optimization process, all randomly initialized neural networks behave quite differently, producing an overdispersed ensemble. After only five to ten iterations, all fifty MLPs mimic better the target data. This behaviour of the individual networks is also illustrated in Fig. 4, which shows the mean error as a function of the training epoch for ten neural networks.

The evolution of the logarithmic score with regard to the number of training epochs is considerably different from the behaviour of the CRPS and of the MAE. After an initial decrease, the logarithmic score increases with the number of training epochs performed because, as the accuracy of the forecasts improves, the corresponding fitted pdf gets narrower, increasing the number of observations falling outside of the pdf. This is confirmed by Fig. 5 in which the occurrence of a small number of daily logarithmic scores above 25, 50 and 100 concurs with the increase in the logarithmic score values. The behaviour of the mean logarithmic score becomes similar to the behaviour of the mean CRPS when only 75 or 90% of the sorted daily scores is used for its computation (Fig. 5b).

The exercise was repeated for the CRPS and the outcome is shown in Fig. 6. First, Fig. 6a shows that the occurrence of large daily CRPS values (above 0.75, 1 and 2) does not vary much during training. However, a minimum is noted around epoch five (for Le Golo River), which corresponds approximately to the number of epochs where the reliability component of the CRPS is minimized (Fig. 7) and where the logarithmic score is minimal (see Fig. 3). Figure 6b shows that the computation of the mean CRPS, contrarily to the mean



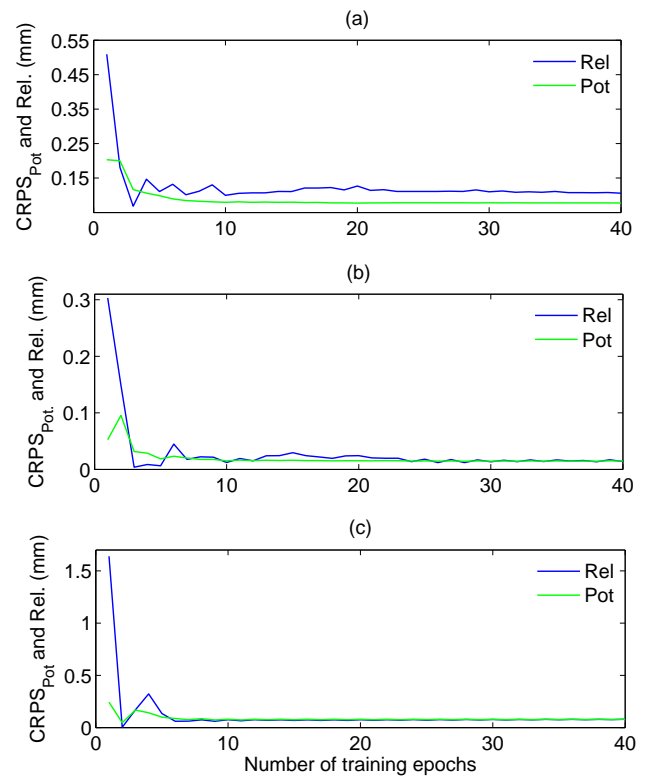
**Fig. 5.** Le Golo River: (a) percentage of logarithmic score values above 25, 50 and 100 as a function of the number of training epochs; (b) mean logarithmic score computed using 100%, 90% and 75% of the ordered daily logarithmic scores.



**Fig. 6.** Le Golo River: (a) percentage of CRPS above 0.75, 1 and 2 as a function of the number of training epochs; (b) mean CRPS computed using 100%, 90% and 75% of the ordered daily CRPS.

logarithmic score, does not change much even if it is performed using 75 or 90% of the sorted daily scores. Therefore, the explanation for the difference of behaviour in the evolution of the two mean scores drawn in Fig. 3 is mainly attributed to the fact that the logarithmic score penalizes more severely than the CRPS when an observation falls in the extreme of the distributions and to the selected method of replacement of infinite logarithmic score values.

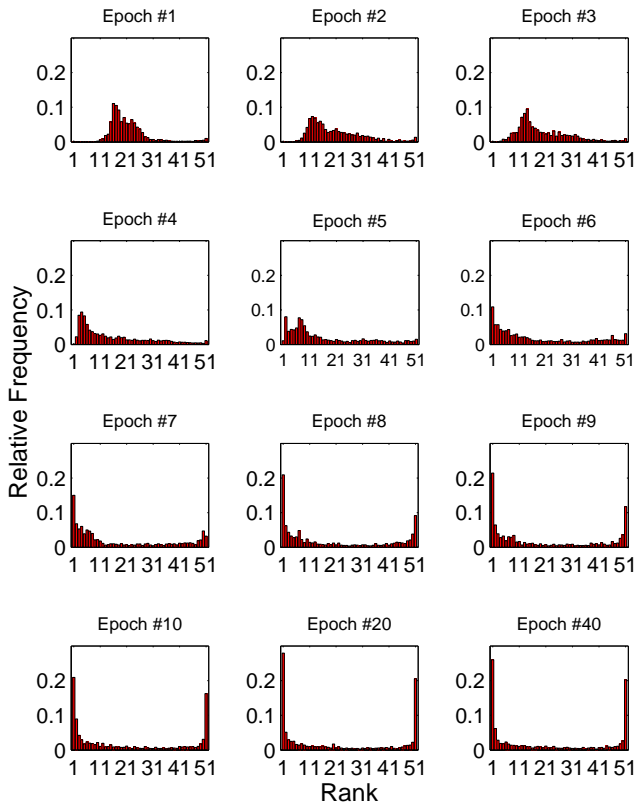
Figure 7 illustrates the evolution of both components of the CRPS with the number of training epochs. For Le Golo and Leaf, both the potential CRPS and the reliability component



**Fig. 7.** Evolution of the Reliability and Potential CRPS with the number of training epochs for (a) Le Golo (b) Serein, and (c) Leaf Rivers.

sharply decrease in the first steps of the training process. Then, the reliability component increases a little, indicating that the ensemble becomes *less* reliable, before stabilizing for the remaining of the training. The Serein River shows a similar pattern for the reliability component, but the potential CRPS initially increases for the first few training epochs before decreasing. Generally speaking, the potential CRPS decreases (i.e. improves) as the training of the networks evolves, which is consistent with the fact that the MLPs turn into more accurate models.

Figure 8 shows the evolution of rank histograms for epoch 1 to 10, 20, and 40 of Le Golo River streamflow forecasts. Rank histograms for the first few iteration of the optimization process are overdispersed. They reflect what is expected from randomly initialized neural networks. The fifty one-day-ahead streamflow forecasts are then very different, so the ensembles exhibit a high variance. Next, as the training continues, each MLP improves and produces more similar forecasts. The spread of the probabilistic distributions is then reduced and the rank histogram flattens (epochs 4 to 7). However, overrepresentation in the lower ranks of the histogram is a probable sign of bias for the testing dataset: the forecasting system often overestimates streamflow. When training continues, all MLPs are converging to the same best

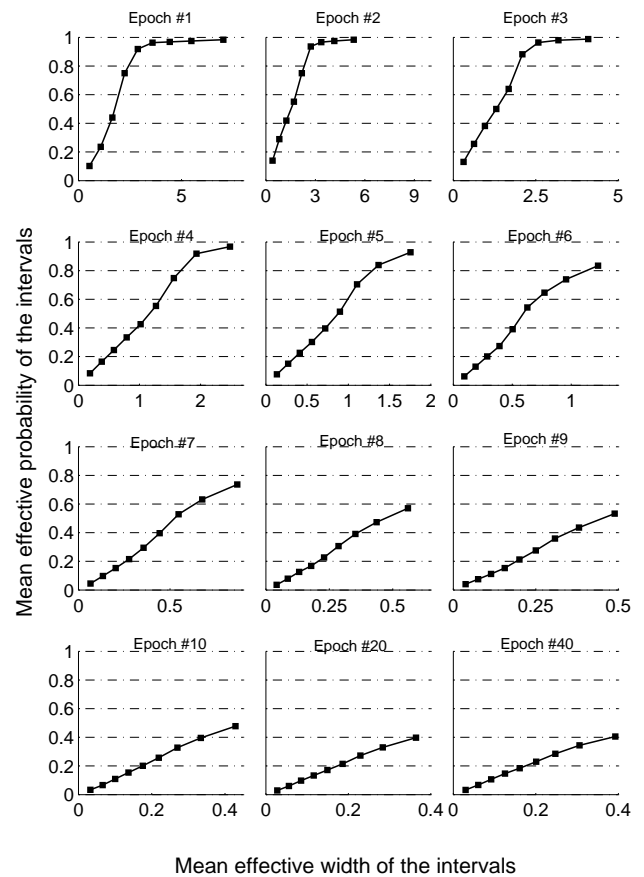


**Fig. 8.** Rank histograms for epochs 1 to 10, 20 and 40 for the Le Golo River.

solution. This leads to sharp but underdispersed distributions, as revealed by an overabundant occurrence of observations ranked first or last in the histograms above epoch 8.

Reliability diagrams, again for Le Golo basin and for epochs 1 to 10, 20, and 40, are drawn in Fig. 9. The y-axis is the average effective coverage of the intervals while the nominal coverage is indicated on the plots for the 0.2, 0.4, 0.6 and 0.8 intervals. The x-axis is the effective width of the intervals. The best situation is when the nominal and effective coverage of the intervals are identical, with the effective width as short as possible (good resolution). On one hand, diagrams for epochs 1 to 4 show effective interval coverage that is greater than the nominal coverage for all confidence intervals, corroborating overdispersed NN ensembles. On the other hand, reliability diagrams for epochs 7 to 10 show the opposite. The most reliable diagrams are obtained at epoch 5 and 6, when effective and nominal coverages almost coincide.

Clearly, this experiment shows that the reliability of confidence intervals, computed from an ensemble of individually trained MLPs, varies as their training progresses. While the accuracy of the forecast computed by averaging the networks forming the ensemble improves with the number of training epochs, the reliability of the confidence intervals may be optimal after only a few training epochs: five to ten for this



**Fig. 9.** Reliability diagrams for epochs 1 to 10, 20 and 40 for the Le Golo River.

experiment. Because this study does not attempt to account for all possible sources of uncertainty (especially the ones linked to the choice of the model architecture), it is not realistic to aim for perfectly reliable confidence intervals. However, we suggest that the results of such an experiment could also be useful to conduct tests on various post-processing methods (e.g. Wilks and Hamill, 2007). All possible situations for raw probabilistic or ensemble forecasts are represented in the results: overdispersion at the beginning of the training, underdispersion at the end, presence of bias to different extent, and some situations where the distribution is almost reliable.

## 5 Conclusions

We have presented an experiment where we trained multiple repetitions of identical MLPs for a one-day-ahead stream-flow forecasting purpose on three watersheds. Instead of focusing on the final performance of the trained networks, we have investigated the properties of these ensembles during their training process. More precisely, we have computed the mean and confidence intervals of the ensembles for each

epoch of the training process. We have assessed the reliability of those intervals and compared the performance of the ensembles with their average value, comparing the MAE and the CRPS. We have also applied the mean logarithmic score and showed that it evolves differently than the mean CRPS as training is performed. Finally, we have also broken the CRPS into its potential and reliability components and showed that its reliability component improves drastically within the first few training epochs: a characteristic that was corroborated by the reliability diagrams and by the rank histograms.

During this experiment, we noted that the CRPS was consistently lower than the MAE, regardless of the number of training epochs. This suggests that it is altogether more advantageous to work with the fifty issued forecasts than to use only their average value. However, because the MLP ensembles constructed here do not account for all possible sources of uncertainties in the streamflow forecasting situation, computed confidence intervals are not reliable, especially after the networks have been trained for more than 10 epochs. Nonetheless, considering the simplicity of implementation of an ensemble of MLPs, especially in contrast with a standard hydrological ensemble prediction system that relies on a complex rainfall-runoff model and on meteorological ensemble forecasts, catchment stakeholders and managers may consider this option as a first order mean to compute close to be reliable short-term streamflow forecasts.

Edited by: E. Toth

## References

- Abrahart, R. J. and See, L.: Comparing neural network and autoregressive moving average technique for the prevision of continuous river flow forecasts in two contrasting catchments, *Hydrol. Process.*, 14, 2157–2172, 2000.
- Anctil, F., Fillion, M., and Tournebize, J.: A neural network experiment on the simulation of daily nitrate-nitrogen and suspended sediment fluxes from a small agricultural catchment, *Ecol. Model.*, 220, 879–887, 2009.
- Baringhaus, L. and Franz, C.: On a new multivariate two-sample test, *J. Multivariate Anal.*, 88, 190–206, 2004.
- Ajami, N., Duan, Q., and Sorooshian, S.: An integrated hydrologic Bayesian multimodel combination framework: Confronting input, parameter, and model structural uncertainty in hydrologic prediction, *Water Resour. Res.*, 43, W01403, doi:10.1029/2005WR004745, 2007.
- Anctil, F. and Lauzon, N.: Generalisation for neural networks through data sampling and training procedures, with applications to streamflow predictions, *Hydrol. Earth Syst. Sci.*, 8, 940–958, 2004, <http://www.hydrol-earth-syst-sci.net/8/940/2004/>.
- Breiman, L.: Bagging predictor, *Mach. Learn.*, 24, 123–140, 1996.
- Campolo, M., Andreussi, P., and Soldati, A.: River flood forecasting with neural network model, *Water Resour. Res.*, 35, 1191–1197, 1999.
- Carney, M., Cunningham, P., Dowling, J., and Lee, C.: Predicting probability distributions for surf height using an ensemble of mixture density networks, 22nd International Conference on Machine Learning, Bonn, Germany, 2005.
- Coulibaly, P., Anctil, F., and Bobbée, B.: Hydrological forecasting with neural networks: The state of the art, *Can. J. Civil Eng.*, 26, 293–304, 1999.
- Crespo, J., Mora, E., and Peire, J.: Estimation Performance of Neural Networks, 531–534, IEEE International Symposium on Industrial Electronics ISIE'93, Budapest, Hungary, 1993.
- Cybenko, G.: Approximation by superposition of a sigmoidal function, *Math. Control Signal.*, 2, 303–314, 1989.
- Efron, B. and Tibshirani, R.: An introduction to the bootstrap, edited by: Cox, D. R., Hinkley, D. V., and Reid, N., Hall, London, 1993.
- Freitas, P. S. A. and Rodrigues, A. J. L.: Model combination in neural-based forecasting, *Eur. J. Oper. Res.*, 173, 801–814, 2006.
- Freund, Y. and Shapire, R. E.: Experiments with a new boosting algorithm, in: Proc. of 13th International Conference on Machine Learning, edited by: Saitta, L., Bari, Italy, Morgan Kaufmann, San Francisco, CA, USA, 148–156, 1996.
- Gneiting, T. and Raftery, A.: Strictly proper scoring rules, prediction, and estimation, *J. Am. Stat. Assoc.*, 102, 359–378, 2007.
- Good, I.-J.: Rational Decisions, *J. Roy. Stat. Soc. B*, 14, 107–114, 1952.
- Hamill, T. and Colucci, S.: Verification of eta-RSM short-range ensemble forecasts, *Mon. Weather Rev.*, 125, 1312–1327, 1997.
- Hansen, L.-K.-H. and Salamon, P.: Neural network ensembles, *IEEE T. Pattern Anal.*, 12, 993–1001, 1990.
- Hersbach, H.: Decomposition of the continuous ranked probability score for ensemble prediction systems, *Weather Forecast.*, 15, 550–570, 2000.
- Hornik, K., Stinchcombe, M., and White, H.: Multilayer feedforward networks are universal approximators, *Neural Networks*, 2, 359–366, 1989.
- Hsieh, W.-W. and Tang, B.: Applying neural network models to prediction and data analysis in meteorology and oceanography, *B. Am. Meteorol. Soc.*, 79, 1855–1870, 1998.
- Iyer, M.-S and Rhinehart, R.-R: A method to determine the required number of neural-network training repetitions, *IEEE T. Neural Networ.*, 10, 427–432, 1999.
- Jolliffe, I. and Stephenson, D.: Forecast Verification: A Practitioner's Guide in Atmospheric Sciences, John Wiley and Sons, Chichester, 2003.
- Kang, K.-W., Park, C.-Y., and Kim, J.-H.: Neural network and its application to rainfall-runoff forecasting, *Korean J. Hydrosoci.*, 4, 1–9, 1993.
- Karunanithi, N., Grenney, W.-J., Whitley, D., and Bovee, K.: Neural networks for river flow prediction, *J. Comput. Civil Eng.*, 8, 201–220, 1994.
- Khan, M. and Coulibaly, P.: Bayesian neural network for rainfall-runoff modeling, *Water Resour. Res.*, 42, W07409, doi:10.1029/2005WR003971, 2006.
- Kingston, G., Lambert, M. F., and Maier, H.: Bayesian training of artificial neural networks used for water resources modeling, *Water Resour. Res.*, 41, W12409, doi:10.1029/2005WR004152, 2005.

- Klemeš, V.: Operational testing of hydrological simulation models, *Hydrolog. Sci. J.*, 31(1), 13–24, 1986.
- Knutti, R., Stocker, T.-F., Joos, F., and Plattner, G.-K.: Probabilistic climate change projections using neural networks, *Clim. Dynam.*, 21, 257–272, 2003.
- Kohonen, T.: The Self-organizing map, *Proc. IEEE*, 79, 1464–1480, 1990.
- Lajbcygier, P.-R. and Connor, J.-T.: Improved option pricing using artificial neural networks and bootstrap methods, *Int. J. Neural Sys.*, 8, 457–471, 1997.
- Levenberg, K.: A Method for the solution of certain non-linear problems in least squares, *Q. Appl. Math.*, 2, 164–168, 1944.
- Mackay, D.: A practical Bayesian framework for backpropagation networks, *Neural Comput.*, 4, 448–472, 1992.
- Maier, H.-R. and Dandy, G.-C.: Neural Networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications, *Environ. Modell. and Softw.*, 15, 101–124, 2000.
- Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.*, 11, 431–441, 1963.
- Neal, R.: *Bayesian learning for neural networks*, Springer, New York, 1996.
- Papadopoulos, G. and Edwards, P.-J.: Confidence estimation methods for neural networks: A practical comparison, *IEEE T. Neural Network.*, 12, 1278–1287, 2001.
- Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain, *Psychol. Rev.*, 65, 386–408, 1958.
- Roulston, M.-S. and Smith, L.-A.: Evaluating probabilistic forecasts using information theory, *Mon. Weather Rev.*, 130, 1653–1660, 2002.
- Shrestha, D.-L. and Solomatine, D.-P.: Experiments with AsaBoost.RT, an improved boosting scheme for regression, *Neural Comput.*, 18(7), 1678–1710, 2006.
- Shrestha, D.-L. and Solomatine, D.-P.: Machine learning approaches for estimation of prediction interval for the model output, *Neural Networks*, 19, 225–235, 2006.
- Singh, P. and Deo, M.: Suitability of different neural networks in daily flow forecasting, *Appl. Soft Comput.*, 7, 968–978, 2007.
- Székely, G.J. and Rizzo, M.L.: A New Test for Multivariate Normality, *J. Multivariate Anal.*, 93, 58–80, 2005.
- Talagrand, O., Vautard, R., and Strauss, B.: Evaluation of probabilistic prediction systems., ECMWF Workshop on Predictability, Shinfield Park, Reading, Berkshire, 1–25, 1997.
- Tokar, A.-S. and Johnson, P.-A.: Rainfall-runoff modeling using artificial neural networks, *J. Hydrol. Eng.*, 4, 232–239, 1999.
- Valverde Ramírez, M.-C., De Campos Velho, H.-F., and Ferreira, N.-J.: Artificial neural network technique for rainfall forecasting applied to the São Paulo region, *J. Hydrol.*, 301(1–4), 146–162, 2005.
- Velázquez, J. A., Petit, T., Lavoie, A., Boucher, M.-A., Turcotte, R., Fortin, V., and Anctil, F.: An evaluation of the Canadian global meteorological ensemble prediction system for short-term hydrological forecasting, *Hydrol. Earth Syst. Sci.*, 13, 2221–2231, 2009, <http://www.hydrol-earth-syst-sci.net/13/2221/2009/>.
- Wilks, D. and Hamill, T.: Comparison of ensemble-MOS methods using GFS reforecasts, *Mon. Weather Rev.*, 135, 2379–2390, 2007.
- Wilks, D.-S.: *Statistical Methods in the Atmospheric Sciences*, Academic Press, San Diego, 1995.
- Zhang, X., Liang, F., Srinivasan, R., and Van Liev, M.: Estimating uncertainty of streamflow simulation using Bayesian neural networks, *Water Resour. Res.*, 45, W02403, doi:10.1029/2008WR00730, 2009.