An Objective Method to Modify Numerical Model Forecasts with Newly Given Weather Data Using an Artificial Neural Network

Ko Koizumi

Meteorological Research Institute, Nagamine, Tsukuba, Ibaraki, Japan

(Manuscript received 3 November 1997, in final form 24 September 1998)

ABSTRACT

An objective method of forecasting precipitation coverage with a neural network is presented. This method uses as predictors all available data at local weather stations including both numerical model results and weather data obtained later than the model initial time, which sometimes contradict each other and hence have to be handled subjectively by well-experienced forecasters. Since the method gives an objective and also realistic forecast of areal precipitation coverage, its skill scores are better than those of the persistence forecast (after 3 h), the linear regression forecasts, and numerical model precipitation prediction.

1. Introduction

In operational forecasting in Japan, forecasters are provided numerical model guidance products twice a day, and they make forecasts based on these products. However, the model predictions sometimes become inconsistent with the actual weather, and hence the model results must be modified according to the actual weather. The modification is a difficult task even for expert forecasters because dependence on the actual weather varies case by case according to the weather condition, which is very difficult when the space–time resolution of the forecast is high.

An artificial neural network is a helpful tool because of its ability to approximate an uncertain function that relates forecast data to the actual weather. The main benefit of using a neural network is that it can account for nonlinear relationships between predictors and predictand. And also, unlike the multiple regression method, which is often used in applying the model output statistics method, the neural network might have a large number of predictors. Indeed, neural networks are less prone to overfitting data than many other models (Bishop 1996). Therefore, various kinds of predictors can be used, including numerical model results and recent observations. Moreover, information from a large extent of the atmosphere can be extracted by using values from a number of grid points that are not necessarily near the forecast area.

The neural network technique has already been applied to prediction of various weather events (McCann 1992; Marzban and Stumpf 1996, 1998). For example, McCann made forecasts of severe thunderstorms using a neural network. His neural network used the lifted index and moisture convergence as input and gave a value between 0 and 1 corresponding to nonoccurrence and occurrence of thunderstorms, respectively. He showed that the critical success index of human thunderstorm forecasts improved from 0.17 to 0.22 when provided with the neural network output. The neural network of McCann used only observational data as input. At the Japan Meteorological Agency (JMA), on the other hand, neural networks are used as substitutes for multiple linear regression traditionally used in model output statistics applications. The predictors used in the networks of JMA are the same ones that were used in the multiple regression method. They are taken only from numerical model products and are ‘‘translated’’ to weather forecasts. To date, there has been no neural network application to weather forecasting that tries to use both numerical model results and observational data.

In the present research, a neural network technique is applied to make precipitation coverage forecasts using all available data including both numerical model output and weather data obtained later than the initial time of numerical weather prediction (NWP) model. Its results can be used as a basis for decision making by forecasters. Section 2 describes the structure and the learning procedure of the network. Section 3 provides the evaluation of the network. Section 4 gives concluding remarks.

2. Neural network configuration

a. Structure of the network

The neural network in the present research has a three-layer structure: input layer, hidden layer, and output layer.

Corresponding author address: K. Koizumi, Meteorological Research Institute, Nagamine, Tsukuba, Ibaraki 305-0052, Japan.
E-mail: kkoizumi@mri-jma.go.jp

All available data that seem to be related to precipitation over the forecast area were included in the predictors. The numerical models used in the present research are the Asia Spectral Model (ASM) and the Japan Spectral Model (JSM) of the Japan Meteorological Agency; both models were used operationally when the research of the present paper was done (the operational NWP model was renovated in March 1996 and the new Regional Spectral Model has been running since then). ASM and JSM ran twice a day: the initial times were 0000 and 1200 UTC. And their products were delivered to local weather stations in the form of gridpoint values.

From ASM, divergence of **Q** vector (Hoskins et al. 1978) at the 500-hPa level, gradient of equivalent thickness (Huber-Pock and Kress 1989), and relative humidity calculated from equivalent thickness (HIX in Huber-Pock and Kress 1989) were input to the network. From JSM, the gridpoint values of vertical velocity at the 700-hPa level, *u* and *v* components of wind at the 850-hPa level, total cloud amount, middle-level cloud amount, and 3-h precipitation amount were put to the network and HIX, Showalter's stability index, temperature advection at the 500-hPa level, and divergence of water vapor flux at 900 hPa were calculated from the gridpoint values of height, wind, temperature, and dewpoint depression at 500-, 700-, 850-, and 900-hPa levels and used as predictors.

Twenty-five grid points of the ASM and 117 grid points of the JSM were selected to cover the forecast area. Nine-point averages of the JSM gridpoint values were made in order to reduce the amount of input data. The numerical prediction data that were valid at the beginning of the specified forecast time were used together with the data that were valid at the end of the forecast time. The number of predictors is 150 for ASM (three kinds of data × 25 grid points × two valid times) and 260 for JSM (10 kinds of data × 13 nine-point averages × two valid times).

From observational data, temperature difference between surface observation and 850-hPa forecast of JSM, *u* and *v* components of surface wind, divergence of surface wind, the radar-observed precipitation amount calibrated with rain gauges (Makihara 1996; Makihara et al. 1996), and satellite-observed infrared imagery were used.

The surface wind and temperature observations are from the Automated Meteorological Data Acquisition System (AMeDAS) interpolated to ¼° lat–long grid points. The interpolation method is based on Barnes (1973), which makes gridpoint value as follows:

$$x_g = \frac{\sum\limits_{R\le R*} x_o \exp(-\alpha R^2)}{\sum\limits_{R\le R*} \exp(-\alpha R^2)},$$

where x_g is the gridpoint value, x_o the observed value, and *R* the distance between the grid point and the observation point. The summation is made for the observation of which the distance is less than *R**. The *R** was set to be the same as the grid distance in the present research. Here, α is the smoothness parameter. The smaller the α is, the smoother the interpolated field becomes. In the present research, the α was set to be 0.4 after some trials. Locations of AMeDAS and the grid points are shown in Fig. 1. Forty-two grid points are taken and hence the number of predictors from AMeDAS is 168 (four kinds of data × 42 grid points).

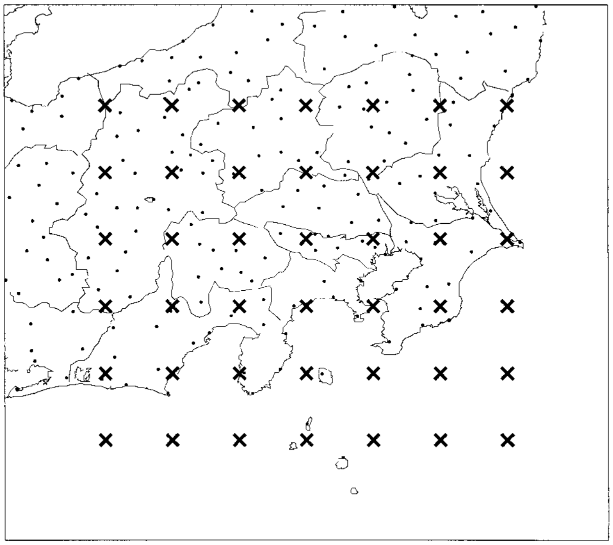


FIG. 1. Location of AMeDAS observation points (shown with dots). Observed data are interpolated to make gridpoint values, which are given to the neural network. Positions of the grid points are marked with ×'s.

The radar-observed precipitation amount and satellite-observed infrared imagery are averaged over the 20 areas shown in Figs. 2 and 3, respectively. In order to take into account seasonal change, sin($2\pi T$/365) and cos($2\pi T$/365) were included in the predictors, where *T* is the Julian date. The total number of predictors including a constant is 621 (150 + 260 + 168 + 20 + 20 + 2 + 1).

All input values are transformed as follows to have a value between 0 and 1[1]:

$$\hat{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}},$$

where *x* is the predictor, $x_{\max}$ the maximum value, and $x_{\min}$ the minimum value of *x* in the training dataset and $\hat{x}$ is the input value to the network.

The predictors have six types of frequency distribution. Surface wind, divergence of surface wind, temperature difference between surface observation and

[1] This is one of various scaling methods. Many neural network developers prefer *z* scores to this method because the training becomes more stable (cf. Bishop 1996).

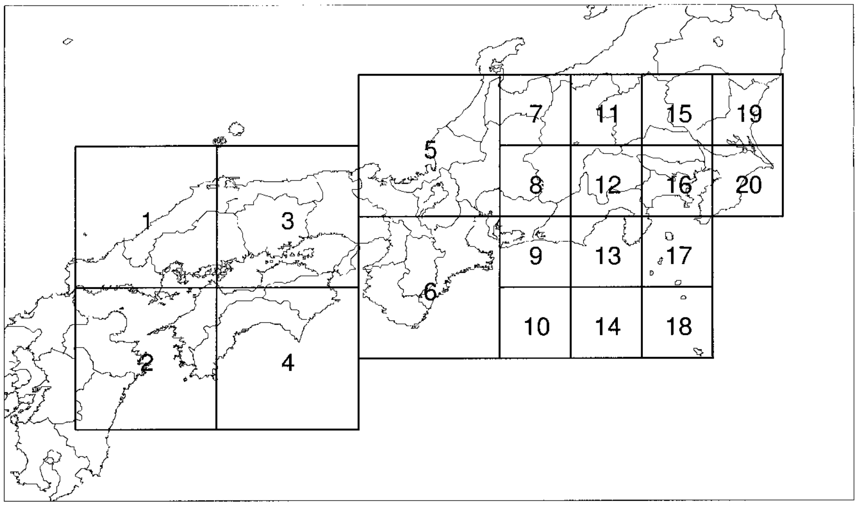


FIG. 2. Radar-observed precipitation amounts averaged over each box and given to the neural network as input.

850-hPa JSM forecast, and wind at the 850-hPa level of JSM have a Gaussian type of distribution (Fig. 4a). Divergence of **Q** vector, temperature advection at the 500-hPa level, vertical velocity at the 700-hPa level, and divergence of water vapor flux at the 900-hPa level have distributions with a steep peak (Fig. 4b). Distribution of HIX and Showalter's stability index have no definite peaks of frequency (Fig. 4c). Distribution of gradient of equivalent thickness and satellite imagery is biased to the lower side of the value range (Fig. 4d). Total cloud amount and the middle-level cloud amount of JSM have high frequency at 1.0 and relatively high frequency at 0.0 (Fig. 4e). Radar-observed precipitation amount and JSM precipitation forecast have definitely high frequency at 0.0 and the frequency decreases along the value increase (Fig. 4f).

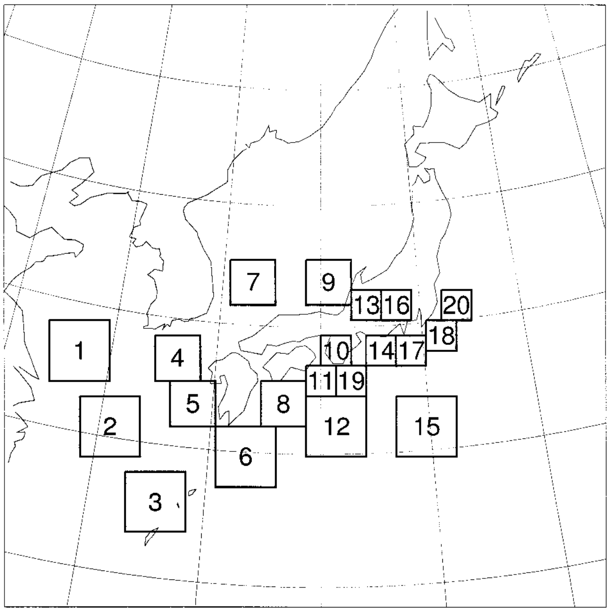


FIG. 3. Pixel values (64 ranked) of *GMS-4* infrared low-resolution fax image averaged over each box and given to the neural network as input. Latitude–longitude lines and coast lines are superimposed on the fax image; therefore, areas were selected to exclude as many of those lines as possible.

Although some predictors (especially gridpoint values of neighboring grids) are highly correlated, no steps were taken to handle the collinearity.

The hidden layer of the network has 200 neurons. The number of hidden neurons is related to the memory of the neural network; that is, the network can memorize more patterns if it has more hidden neurons. On the other hand, however, the network tends to ''overfit'' to noisy data when it has too many hidden neurons. Hence it is crucial to set a ''proper'' number of hidden neurons. In the present research, the number of hidden neurons was set to be maximum within the limit of the computational resources and at the same time an algorithm to avoid overfitting was employed.

The output layer has 120 neurons corresponding to precipitation forecasts for each of the 120 areas shown in Fig. 5. As a predictand, the ratio of radar-observed rainfall area to forecast area is employed.

Each hidden and output neuron takes the linear combination of all values of the previous layer and transforms it with the sigmoid function $1/[1 + \exp(-x)]$, and gives the result to the next layer (Fig. 6). The weights of the linear combination are adjusted in the course of training in order to simulate the input–output relationship of the given training data. This involves the minimization of an error function, which in this article is taken to be the sum of the square error.

The multiple linear regression method was also applied to the same training data for comparison. For each one of 120 forecast areas, one regression function was built. Since the predictors in the present research are

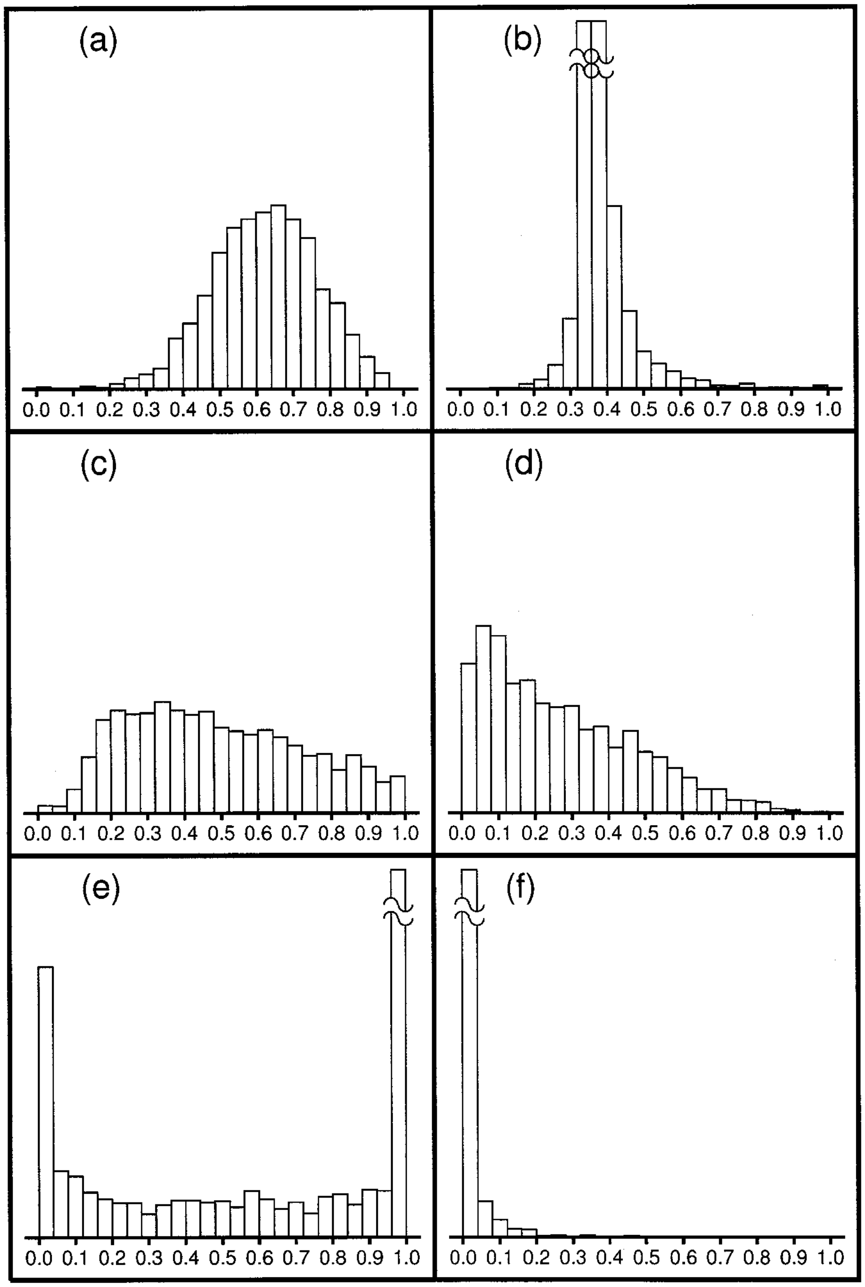
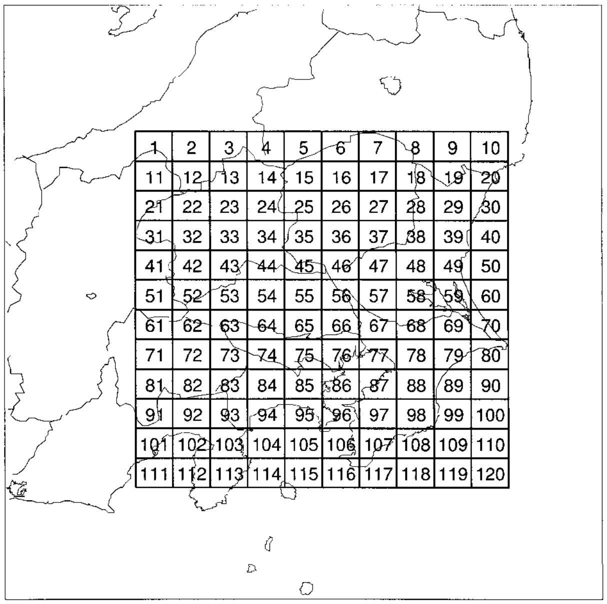


FIG. 4. Histograms of (a) divergence of surface wind, (b) divergence of **Q** vector, (c) HIX of ASM, (d) pixel value of satellite-observed infrared imagery, (e) total cloud amount of JSM, and (f) radar-observed precipitation amount.

prepared for all 120 area forecasts, it might not be necessary for a regression function to use all the predictors. Hence, the stepwise algorithm was employed to select only necessary predictors for each regression function. *F* values of the stepwise algorithm were set to be 10.0 for acceptance of a predictor and 7.0 for removal of a predictor after some trials. The number of accepted predictors was between 13 and 40.

Although the selection of the ''best'' predictors might improve the performance of a neural network, the predictors chosen by the stepwise analysis are not always the ''best'' for the neural network, because the stepwise method evaluates the importance of predictor only with linear relationship between the predictor and predictand while neural networks can reproduce both linear and nonlinear relationships. Little is known of how to choose the ''best'' predictors for a neural network before training it.



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

FIG. 5. Forecast areas. The neural network was designed to forecast whether each area will have precipitation or not.

b. Learning procedure of the network

In the present research four neural networks were constructed: one neural network is for 0–3-h forecast, one for 3–6-h forecast, one for 6–9-h forecast, and one for 9–12-h forecast. Each network is supposed to make a forecast every 3-h. For example, given observational data of 0900 UTC, the network for 6–9-h forecasts makes a forecast of 120 areas' precipitation coverage during 1500–1800 UTC using observational data of 0900 UTC and the numerical model results of 0000 UTC initial, which are valid at 1500 and 1800 UTC. If the time of observation is between 0300 and 1200 UTC, the numerical model results of 0000 UTC initial are used and otherwise those of 1200 UTC initial are used.

Four datasets were constructed for training of each network using the results of numerical models and observational data from March 1994 to February 1995. The number of samples in each set is 2679, 2682, 2676, and 2678 for 0–3-, 3–6-, 6–9-, and 9–12-h forecasts, respectively. The numbers are smaller than the supposed 2920 (eight times a day for 365 days) because samples lacking some of its components were removed from the set.

Among the various kinds of training algorithms, the conjugate-gradients method (Fletcher and Reeves 1964) was chosen, since it converges faster to a minimum value than the backpropagation algorithm and is applicable to a large-size neural network. In order to minimize a given function, the conjugate-gradients method makes a search for a minimum point along the ‘‘conjugate-gradient direction.'' When a minimum point is found, the conjugate-gradient direction at that point is calculated and a search is made again from the point along the new direction. The above procedure is repeated until the derivative of the function becomes sufficiently small. About 200 iterations were taken in the present research. While conjugate-gradient direction can be calculated in several ways, Fletcher and Reeves proposed one described as follows:

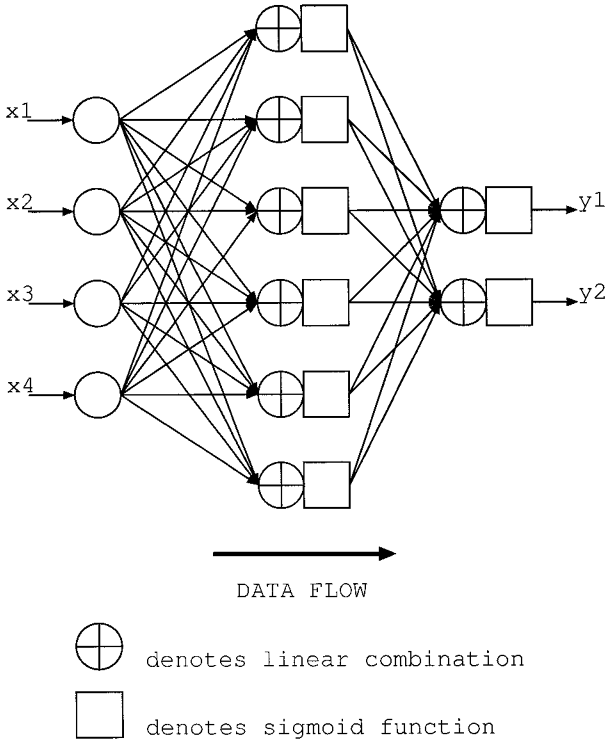


FIG. 6. Schematic figure of a feed-forward neural network with four input nodes and two output nodes, with six hidden nodes on one hidden layer.

$$d_k = -\nabla F(x_k) + \frac{\|\nabla F(x_k)\|^2}{\|\nabla F(x_{k-1})\|^2} d_{k-1},$$

where F is the function to be minimized, d_k the direction of the kth search, and x_k the minimum point found at the $(k - 1)$-th search. Usually, d_1 is given as $-\nabla F(x)$.

The weight decay method (e.g., Bishop 1996) is also introduced into the training algorithm. The network in the present paper has 148 320 parameters to be adjusted, 124 200 (=621 × 200) between the input and hidden layers, and 24 120 (=201 × 120) between the hidden and output layers, and such a large network tends to *overfit* the training data and loses generalization because the number of parameters to fit is much larger than the number of cases in the developmental sample. The weight–decay method is a way to avoid overfitting, which adds a penalty term to the error function as follows:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{120} [f_j(\mathbf{x}_i) - \mathbf{y}_{ij}]^2 + \eta \sum_{k=1}^{148\,320} w_k^2,$$

where E is the error function, N the number of training samples, f_j the network output for the jth predictand, $\mathbf{x}_i$ the input vector of the ith sample, $\mathbf{y}_{ij}$ the true output of the jth predictand of the ith sample, and w the connection weights in the neural network. The penalty term, constraining the size of weights, prevents the network from overfitting to the given samples, for, with large parameter values, the sigmoid function becomes very steep and has strong nonlinearity as a result the whole neural network represents a very nonlinear function. The positive value η controls the extent of constraint. When η is too large, the weights are strongly constrained and the neural network becomes a too smooth function and fails to approximate the input–output relationship of samples properly. With too small η, on the contrary, the neural network tends to fit to a noisy detail of the input–output relationship of samples. Hence, to select a proper value of η is crucial. Although a method for the selection of η value is presented with Bayesian theory (MacKay 1995), it was not employed in the present research because the method requires several iterations of all the training procedure and it takes too much computational time, which is not affordable at this time. The value η was set to be 0.1 in the present research in order that the two terms of the error function are comparable when an average size of error $|f_j(x_i) - y_{ij}|$ is around 0.2 and an average size of parameter $|w^2|$ is around 1.0. As the value is an arbitrary one, other values of η should be tested for improving the performance of the network in the future.

3. The network's skill

The neural network in the present research was designed to be a tool to make rain/no-rain forecasts for 120 forecast areas. The network's output equal to or above a threshold value θ is taken as the ''rain'' forecast and output below θ is taken as the ''no rain'' forecast for a specified forecast area. One threshold value is used for all the forecast areas in one month and the value varies month to month. The value θ was so defined as to maximize the Heidke skill score (HSS) in the training dataset.

The skill of the network is evaluated with HSS calculated as follows:

$$S = \frac{T - C}{N - C},$$

where S is the skill score, T the number of hits, N the total number of forecasts, and C the number of climatological hits. The value of C is calculated as

$$C = \frac{F_1 \times O_1 + F_0 \times O_0}{N},$$

where F_1 and F_0 are the number of rain forecasts and no-rain forecasts, and O_1 and O_0 are the number of ''actual rain'' and no rain. The score becomes 1.0 for the perfect forecasts and 0.0 for random or climatological forecasts. The two-by-two contingency table was made for each month using forecast results of all 120 areas. Hence N is around 28 800 (8 × 30 × 120). These scores were calculated for the period between March 1995 and February 1996, which did not overlap with the training period. The number of samples in this period is 2731, 2727, 2722, and 2716 for the 0–3-, 3–6-, 6–9-h, and 9–12-h forecasts, respectively. Samples lacking some of its components were removed from the set in the same way as of the training set. The monthly precipitation appearance rate (O_1/N) during the period varied from 0.04 (December 1995) to 0.37 (July 1995).

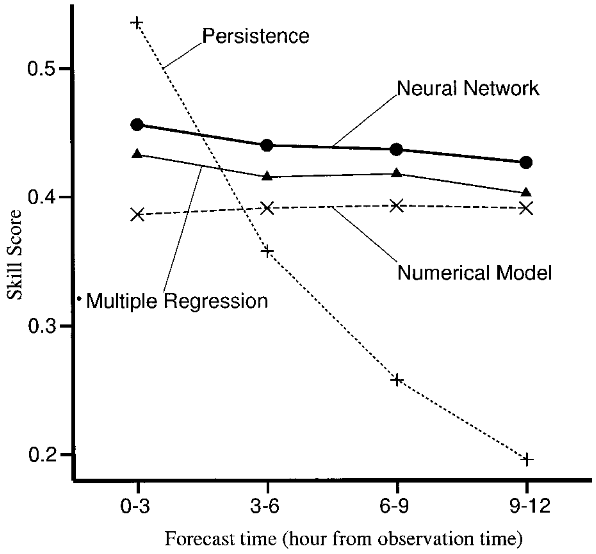


FIG. 7. Skill scores of the neural network forecasts, the multiple linear regression, JSM prediction, and persistence forecast according to lag from the observation time. The skill scores are calculated monthly from Mar 1995 to Feb 1996, and 12-month averages are made.

It should be noted that training a neural network with 1 yr of data and evaluating it with another 1 yr of data cannot give a nonbiased generalization performance of the network. Hence the following results provide limited information about the performance of the neural network, as they are based on 1-yr validation, which may have some biases.

Figure 7 shows 12-month averages of monthly skill scores of the neural network, JSM, multiple linear regression method, and persistence forecast. In order to make rain/no-rain forecasts from JSM, the 3-h precipitation amount of JSM at the grid point nearest to the forecast area was taken. Both the JSM gridpoint value and the multiple linear regression result were dichotomized into two categories (i.e., rain forecast and no-rain forecast) using a threshold value defined for each month in the same way as that for the neural network. The ''persistence forecast'' makes the rain forecast on the areas that have 50% or over 50% precipitation coverage at present and the no-rain forecast on the other areas.

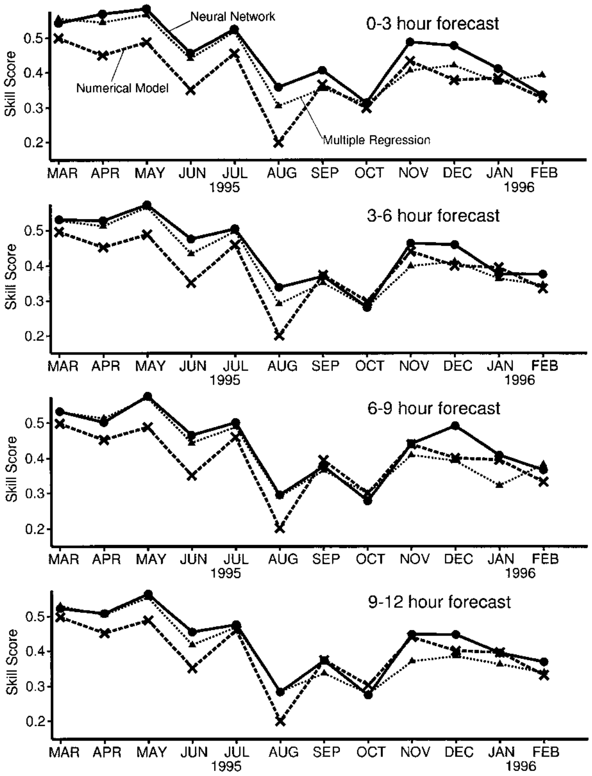


FIG. 8. Monthly skill scores of the neural network forecasts (solid line), the multiple linear regression (dotted line), and JSM prediction (dashed line). From top to bottom, scores of 0–3-, 3–6-, 6–9-, and 9–12-h forecast are shown.

The lead time shown in Fig. 7 is the duration from the time of observation to the forecast time. Therefore, in the validation set of each lead time, the JSM forecast of various forecast times is mixed up and hence the scores of JSM are almost the same for each lead time.

Though the persistence forecast has a high score for the 0–3-h forecast, the neural network is the best of all four methods beyond 3 h.[2]

Figure 8 shows month-to-month skill scores of the neural network, JSM, and the multiple linear regression method. Although the scores are varying month to month, the scores of the neural network are the highest of the three methods except for a few months. JSM had difficulty in forecasting precipitation in August because the precipitation in this month is often brought by small-scale convective systems, which the numerical models can hardly predict. The neural network made better forecasts, even in this month. This may be due to the use of information from the observational data. The difference of the skill scores between the neural network and JSM is reduced according to the forecast lead time. The scores of the multiple regression are as high as the neural network from March 1995 to October 1995 while it falls behind the neural network's from November 1995 to February 1996. It may be because the relationship between predictors and predictands varies season to season. As the multiple linear regression makes an approximation of an ''average'' relationship, it may work well in some seasons and not so well in others. The neural network might be able to make a proper approximation of such relationships with its nonlinearity.

Spatial distribution of 12-month-average skill score is shown in Fig. 9. The neural network's performance shows moderate fluctuation within the region compared to that of JSM, which has, for example, low scores along the east end of the region.

Figure 10 shows a sample of the neural network forecasts compared with JSM's precipitation prediction and the actual precipitation area. The neural network forecasts precipitation over almost the same area as the observations at 0–3 and 6–9 h, while JSM predicts a much smaller precipitation area at 0–3 and 3–6 h and a much larger one at 6–9 h. For the 9–12-h forecast, however, the neural network forecasts nearly the same precipitation area as the JSM prediction, although the actual precipitation covers a larger area than the forecasts. As shown in this case, the neural network seems to be able to make rain forecasts by taking information from observational data even when the numerical model makes no precipitation at all. And it also seems to be able to combine smoothly the latest observation and numerical model results by applying different weights to the latest weather data according to the forecast time.

4. Concluding remarks

In the present paper, an artificial neural network was applied to precipitation coverage forecasts.

The neural network was used in a somewhat ''crude'' manner in the present research; that is, all possible predictors were given to the network without any selection procedure, the number of hidden neurons and the weight–decay parameter were arbitrarily given, and the size of the network was a little too large compared with the volume of the training dataset. Nevertheless, after being trained with the conjugate-gradient method and the weight–decay algorithm, it performs better than the raw use of the numerical model prediction results or the multiple linear regression method. This crude use of a neural network is useful when it is difficult to spare enough time or computational resources for selecting the optimal statistical model or for making a detailed examination of a large amount of predictor candidates.

As the period of the training data is only 1 yr in the present research, the performance of the neural network will be improved when more training data become available.

It is still unclear to what extent each predictor con-

[2] Estimation of error range of the scores requires a large amount of computational resources, which are not affordable at present. Hence the superiority of the neural network is not yet proven to any degree of statistical significance.

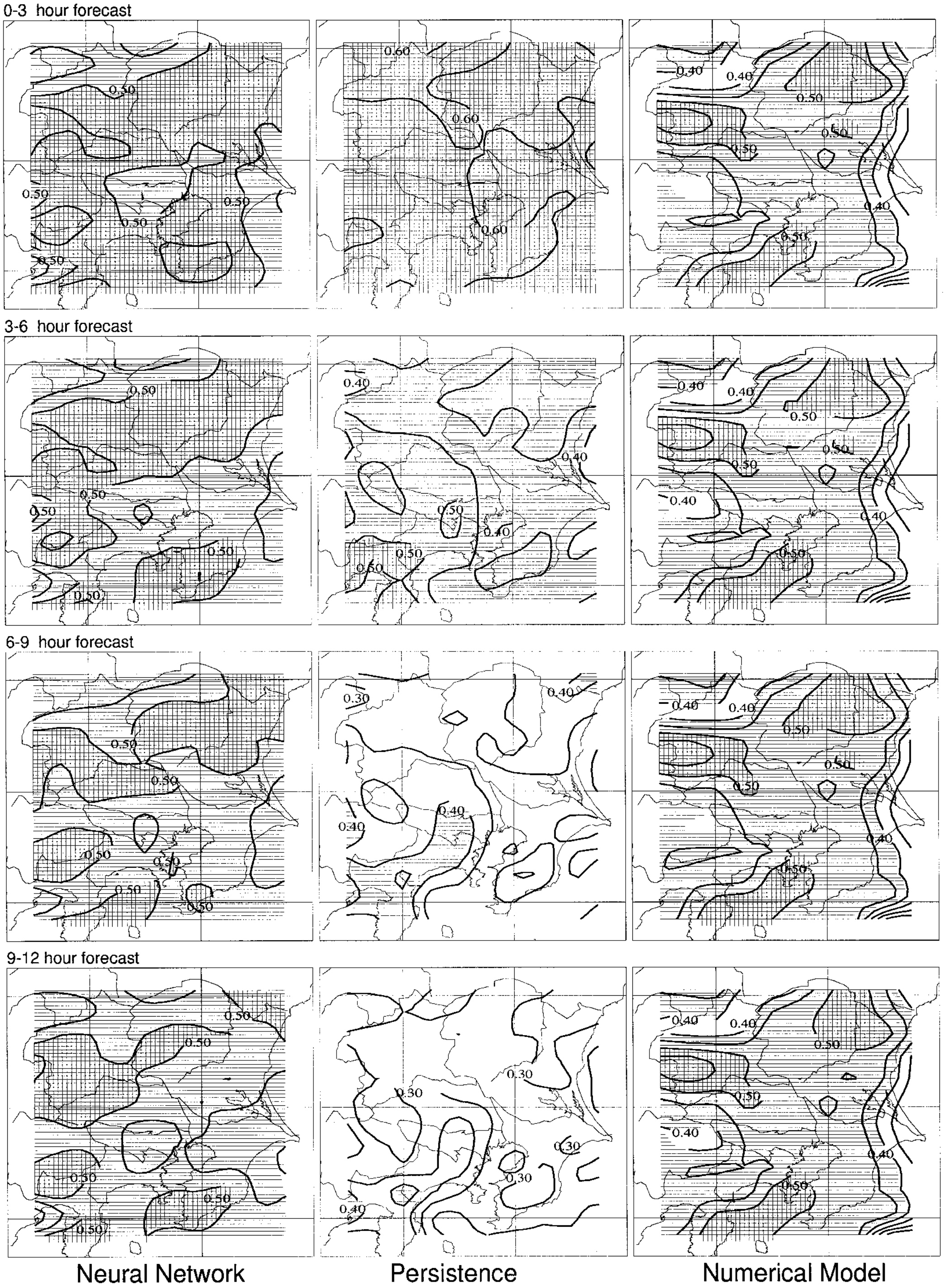


FIG. 9. Spatial distribution of skill score of the neural network (left column), persistence forecast (middle column), and JSM prediction (right column). From top to bottom, scores of 0–3-, 3–6-, 6–9-, and 9–12-h forecast are shown. Areas of score over 0.4 are hatched with horizontal lines and areas of score over 0.5 are hatched with crossing lines. The skill scores are calculated using all the evaluation data from March 1995 to February 1996.

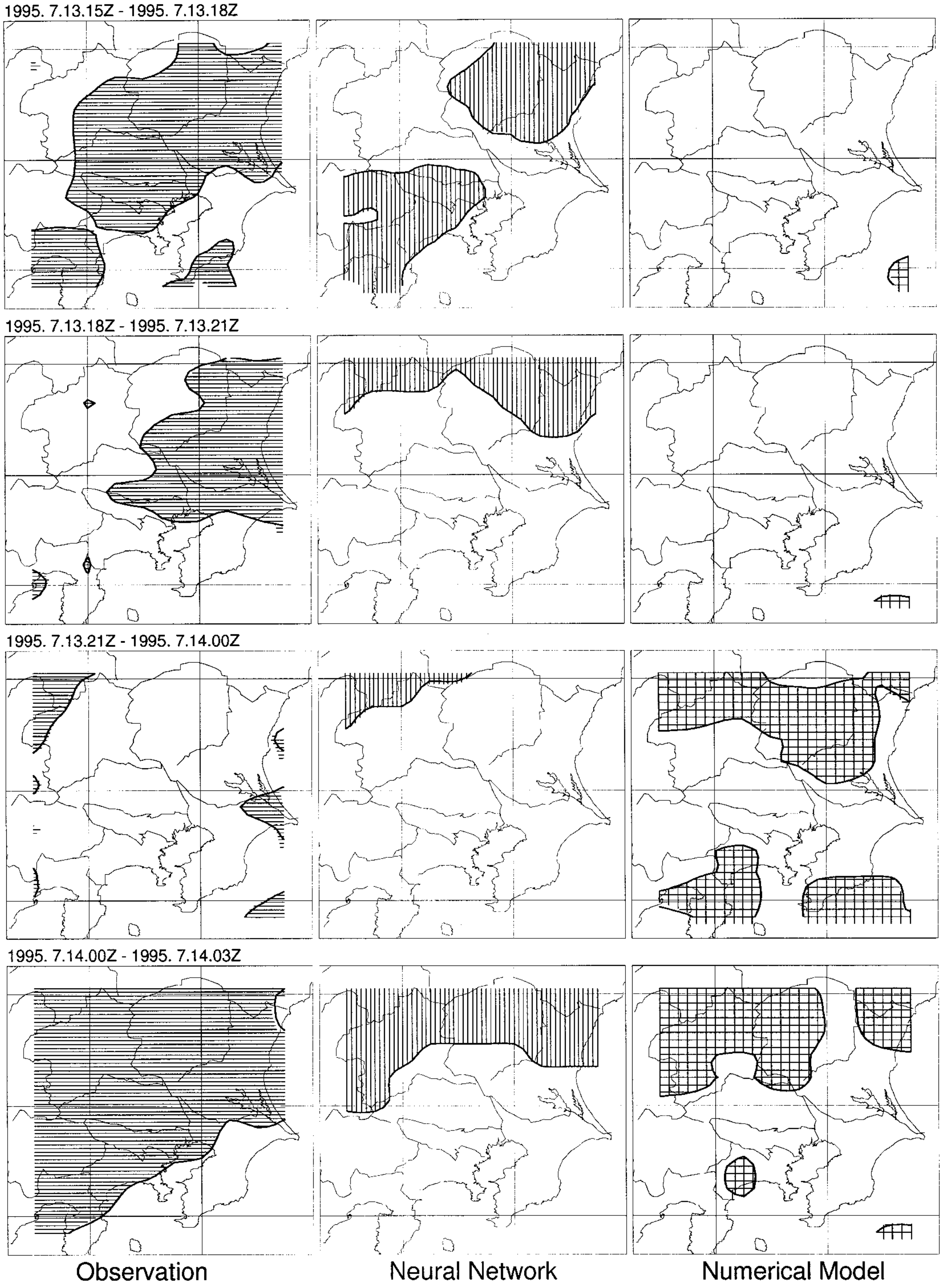

1995. 7.13.15Z - 1995. 7.13.18Z
1995. 7.13.18Z - 1995. 7.13.21Z
1995. 7.13.21Z - 1995. 7.14.00Z
1995. 7.14.00Z - 1995. 7.14.03Z
Observation
Neural Network
Numerical Model

FIG. 10. (Left column) Areas of precipitation amount over 0.4 mm, which is observed with the radar–AMeDAS composite. From top to bottom, the valid times of every row are 1500–1800 UTC, 1800–2100 UTC, 2100–2400 UTC 13 July 1995, and 0000–0300 UTC 14 July 1995. (Middle column) The neural network's forecasts. Forecasts were made when observational data at 1500 UTC 13 July 1995 were given. Areas of network output over 0.5 are hatched. (Right column) JSM precipitation prediction. Initial time is 0000 UTC 13 July 1995. Areas of 3-h precipitation amount over 0.4 mm are hatched.

tributes to the forecasts and to what extent recent observations improve the forecasts. It could be resolved by constructing a neural network using only a subset of the predictors, for example, only the numerical model results, which should be tried in the future.

There is a forecasting ''gap'' in forecast time of 3–12 h, as Doswell (1986) described. In the gap, linear extrapolation becomes meaningless and numerical models are still in an ''adjustment stage.'' The neural network can be one way to fill up the gap and can support forecasters' tasks.

Acknowledgments. The author wishes to express his thanks to Mr. Yasutaka Makihara in JMA and three reviewers for their helpful comments.

Computations were made on the Hitachi 3050RX/205 Workstation of the Meteorological Research Institute.

REFERENCES

Barnes, S. L., 1973: Mesoscale objective map analysis using weighted time-series observations. NOAA Tech. Memo. ERLTM-NSSL-62, 60 pp. [NTIS COM-73-10781.]

Bishop, C. M., 1996: *Neural Networks for Pattern Recognition.* Clarendon Press, 482 pp.

Doswell, C. A., III, 1986: Short-range forecasting. *Mesoscale Meteorology and Forecasting,* P. S. Ray, Ed., Amer. Meteor. Soc., 689–719.

Fletcher, R., and C. M. Reeves, 1964: Function minimization by conjugate gradients. *Comput. J.,* **7,** 149–154.

Hoskins, B. J., I. Draghici, and H. C. Davies, 1978: A new look at the ω-equation. *Quart. J. Roy. Meteor. Soc.,* **104,** 31–38.

Huber-Pock, F., and C. Kress, 1989: An operational model of objective frontal analysis based on ECMWF products. *Meteor. Atmos. Phys.,* **40,** 170–180.

MacKay, D. J. C., cited 1995: Probable networks and plausible predictions—A review of practical Bayesian methods for supervised neural networks. [Available online from ftp://wol.ra.phy.cam.ac.uk/pub/www/mackay/network.ps.gz.]

Makihara, Y., 1996: A method for improving radar estimates of precipitation by comparing data from radars and raingauges. *J. Meteor. Soc. Japan,* **74,** 459–480.

——, N. Uekiyo, A. Tabata, and Y. Abe, 1996: Accuracy of Radar-AMeDAS precipitation. *IEICE Trans. Commun.,* **E79-B,** 751–762.

Marzban, C., and G. J. Stumpf, 1996: A neural network for tornado prediction based on Doppler radar-derived attributes. *J. Appl. Meteor.,* **35,** 617–626.

——, and ——, 1998: A neural network for damaging wind prediction. *Wea. Forecasting,* **13,** 151–163.

McCann, D. W., 1992: A neural networks short-term forecast of significant thunderstorms. *Wea. Forecasting,* **7,** 525–534.