交通部中央氣象局委託研究計畫成果報告

用神經網路模式來預報台灣颱風季的雨量

Long-range rainfall forecasts for the Taiwan area during summer using neural network models

計畫類別：□國內　■國外
計畫編號：CWB89-3M-07
執行期間：88 年 8 月 1 日至 89 年 12 月 31 日
計畫主持人：朱寶信
執行單位：Climate System Enterprise

中華民國八十九年十一月

Long-range rainfall forecasts for the Taiwan area during summer using neural network models

Final Report to the Central Weather Bureau
Under Contract Number CWB 89-3M-07

Pao-Shin Chu, Ph.D.
Climate System Enterprise
Honolulu, Hawaii, U.S.A.

November 28, 2000

1. INTRODUTION

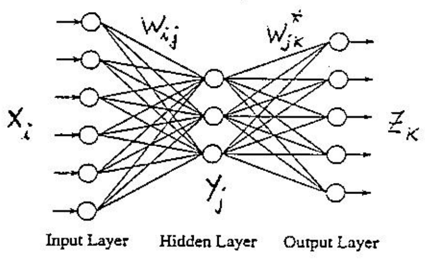
The objective of this project is to develop a prototype neural network (NN) model and test this model's ability in long-lead seasonal rainfall prediction for Taiwan. In my previous projects funded by the Central Weather Bureau, an advanced linear model called canonical correlation analysis (CCA) was developed in predicting Mei-Yu (May/June) and typhoon season (July through October) rainfall for Taiwan (Chu, 1998; Chu and Yan, 2000). Long-term rainfall records (1956-98) from sixteen stations in Taiwan and the antecedent sea surface temperature (SST) data for the Pacific and/or Indian Oceans were used. To provide an overall evaluation of the CCA skill, cross-validated correlation coefficients between forecasts and observations at one-month lead were attempted. The lead time is defined as the number of months between the last month of the predictor (SST) data and the first month of the predictand (rainfall) field. A one-month lead implies that the last month of the SST field is May for the typhoon season prediction. Cross-validation is a computer-intensive method. One point from the SST and rainfall datasets is omitted each time, the CCA model is reconstructed, and forecast for the omitted case (for rainfall) is made. By repeatedly going through the entire records, it is possible to obtain as many sample sizes of the forecast as the original observations.

For the typhoon season prediction, the CCA cross-validated correlation skill at one-month lead reaches 0.43 for Ilan and 0.37 for Keelung (significant at the 5% level). For Taichung, Tainan, and Kaohsiung, however, the skill is near zero or even negative (Chu and Yan, 2000). In general, rainfall predictability is higher in the northeastern Taiwan but much lower (or no skill) for the western plain. To the extent that the Earth's climate system we wish to predict is nonlinear, seasonal rainfall predictability from linear models can only be regarded as a lower bound estimate for the actual predictability of the system.

On the other hand, the NN is a nonlinear and powerful tool which was recently developed to handle a wide range of applications including seasonal climate predictions (e.g., Hastenrath, 1995; Navane and Ceccatto, 1994; Silverman and Dracup, 2000), time series analysis of nonlinear systems such as the Lorenz equations, and statistical downscaling (e.g., Zorita and von Storch, 1999). Basically, a NN is trained to recognize patterns of interests in datasets and this pattern recognition capability can be employed effectively for prediction experiments.

2. NEURAL NETWORK MODELS

We use the fully connected three-layer, feed-forward NN system. In this system, the NN consists of three layers - the input, hidden, and output layers (Fig. 1). Each node



Neural network

Fig. 1. A three-layer feed-forward neural network model. In this example there are six nodes in the input layer, three nodes in the hidden layer, and five neurons in the output layer. The w_{ij} and w^*_{jk} are the weights.

in the input layer brings into the network the value of one independent variable (i.e., X_i). In application, each X_i may represent a dominant empirical orthogonal function (EOF) mode of the SST and the outputs, Z_k, may represent a simulated rainfall at each station in Taiwan. The network is fully connected in the sense that there are links between all the nodes in adjacent layers. Each link has a "weight" and each weight may have different values. Note that a separate link (i.e., different weights) exists between each input node and each hidden node, and between each hidden node and each output node. Often, a bias weight is added to each hidden node and output node.

The network operates in two modes, mapping mode and learning mode. In mapping mode, information flows "forward" from inputs to outputs. Training begins with arbitrary values for the weights and proceeds iteratively. Each iteration is called an epoch. In each epoch, the network adjusts the weights in the direction that reduces the error, which is the difference between the current outputs (e.g., simulated rainfall) and target outputs (e.g., actual rainfall). As the iteration continues, the weights generally converge to an optimal value. Many epochs are required before an optimal value is reached.

In learning mode, information flows "forward" and "backward" consecutively. The input nodes send values to all the hidden nodes. Each hidden node calculates the weighted sum of the inputs. Then each hidden node computes a sigmoid function of its weighted sum. The sigmoid function simply squashes the sum down to a limited range. A variety of sigmoid functions can be used including the hyperbolic tangent function, logistic function, linear transfer function, and others. Subsequently, each hidden node sends the squashed values to all the output nodes. Each output node performs a similar function as the hidden node by calculating the weighted sum from the outputs of the hidden layer and then use a transfer function (linear or nonlinear) to compute new outputs.

For each observation used as an input, there is a forward path through the network to generate the current output. It is followed by a backward path to determine how the weights should be adjusted. At the end of each epoch, all the weights in the hidden and output layers will be changed. The "backpropagation" method is used for adjusting the weights. This name is derived from the fact that error (i.e., the difference between the target function and neural's output) information is sent back from the output nodes to the hidden nodes.

The key in back propagation is the sensitivity of the network's error to changes in its weights. All the nodes change their weights based on the accumulated derivatives of the error with respect to each weight. In other words, it is necessary to determine how the partial derivatives of the error change with respect to the weights. We used a learning law called the steepest descent method, which moves the weights in the direction in which the error declines most rapidly. This is analogous to a skier who always moves downhill through the mountain until he hits the bottom. Steepest descent involves moving a small step down the local gradient of the scalar field. Although this method is a slow technique, it is the only method that has been proven mathematically to converge on the set of weights producing minimum error.

3. PRELIMINARY RESULTS

We accomplished five tasks (1) data collection and processing, and data reduction, (2) development of a forward algorithm, (3) development of an error derivative algorithm, and (4) linkage of various subroutines into a comprehensive computer program. For task (1), we collected and processed seasonal rainfall records for 16 stations in Taiwan. The record length is 1956-1998. All monthly records were kindly provided to us by Mr. G.-H. Chen of the Forecast Center of the CWB. The monthly data were transformed to seasonal values and then standardized. To focus on the dominant modes of variability, the seasonal and standardized rainfall indices are subject to an EOF analysis.

We also successfully collected long-term SST data for the Pacific Ocean (50°N-40°S, 120°E-90°W) and the Indian Ocean (40°S-20°N, 20°E-120°E). Using the original data at 2.5° latitude x 2° longitude grids, these data are then averaged into 10°lat by 10°long gridpoints to facilitate the subsequent analyses. This yields 125 grids for the Pacific Ocean and 41 grids for the Indian Ocean. For the South China Sea, the data set was processed at 4° x 4° because of its smaller domain, yielding 30 grids. Note that the Indian Ocean and the South China Sea datasets are new to this project. By including the Pacific Ocean, the Indian Ocean as well as the South China Sea we can test the variations of nearly global boundary conditions on seasonal rainfall predictibility for Taiwan. Because the SST is known to have persistence, we used the SST data field, evolved over 1-yr to be the predictor variable. Specifically, we calculated the extended EOF (EEOF) at 3-mo intervals over a one year period prior to the rainy season. In addition to spatial compression, the EEOF analysis enables us to capture propagating features of the SST. This is achieved by stacking the four temporal series of the SST field into a large matrix so that the evolution of SST spatial pattern over a one year period is preserved.

As stated in Section 2, the forward procedure (Task 2) goes through each of the hidden nodes in the network and calculates the weighted sum of inputs for each of them. This weighted sum is then fed to a sigmoid function which sends its new value to the output layer. We use a Sshaped sigmoid function so that the sum of all weighted inputs is bounded between 0 and 1. Once the output node receives information from the hidden layer, it will work in the same way as the hidden nodes. During training, its output will be compared to the specified target value. Accordingly, the error derivative procedure (Task 3) calculates the error derivative for each weight.

As an example, Figure 2 shows the time series of a target function and the model's output (Z). The target function is a highly nonlinear sinusoidal wave. In this simple example, only one input node, five hidden nodes, and one output node are considered. A nonlinear logistic function is employed in the hidden layer and a linear transfer function is used in the output layer. The weights are prescribed (fixed) before the experiment. Although not perfect because the changeweights program was not implemented, the simulated output matches the target function quite well.

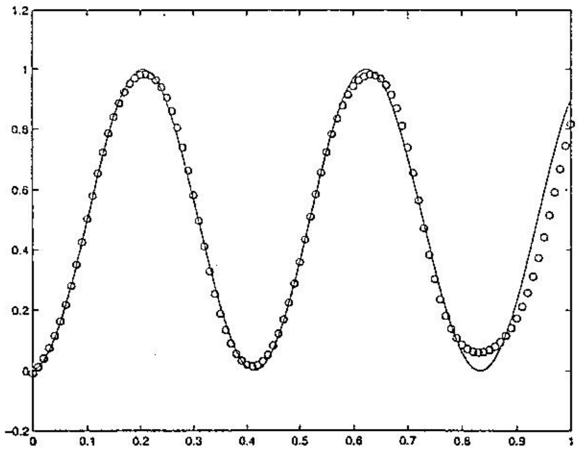


Fig. 2. Target function (solid curve) and network output (open circle) as functions of network input (X).

Since March this year, the changeweights subroutine has been developed and various subroutines have been linked in one computer program (Tasks 3 and 4). We also tested the prototype NN model using a real-world problem. A case in point is the application to rainfall prediction during the typhoon season. In order to appreciate the utility of the NN model, a comparison is made with the CCA prediction. Figure. 3 shows the CCA cross-validated

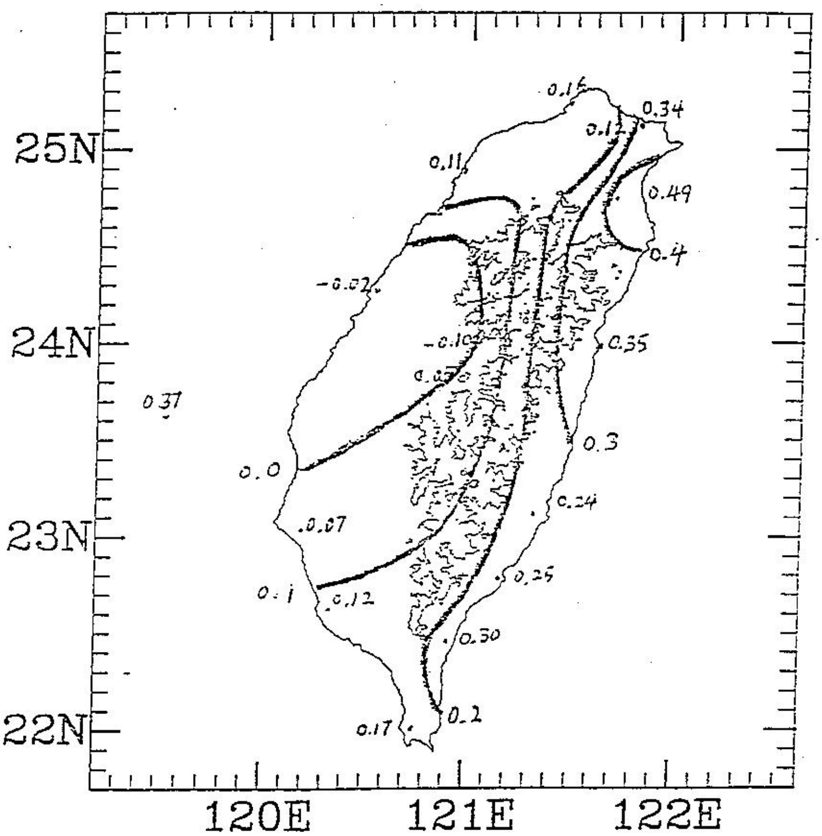


Fig. 3. CCA cross-validation correlation skill for summer (JASO) rainfall at one-month lead.

correlation skill of JASO rainfall at one-month lead for Taiwan. The inputs are the leading eight modes of the SST field and the leading four modes of the rainfall field. These modes explain 85% of the variance in the SST datasets and 60% of the variance in the rainfall datasets.

Figure 4 shows the similar skill at one-month lead but using the NN model. In comparison to Fig. 3, the skill for the northeastern Taiwan from the NN model remains essentially the same as

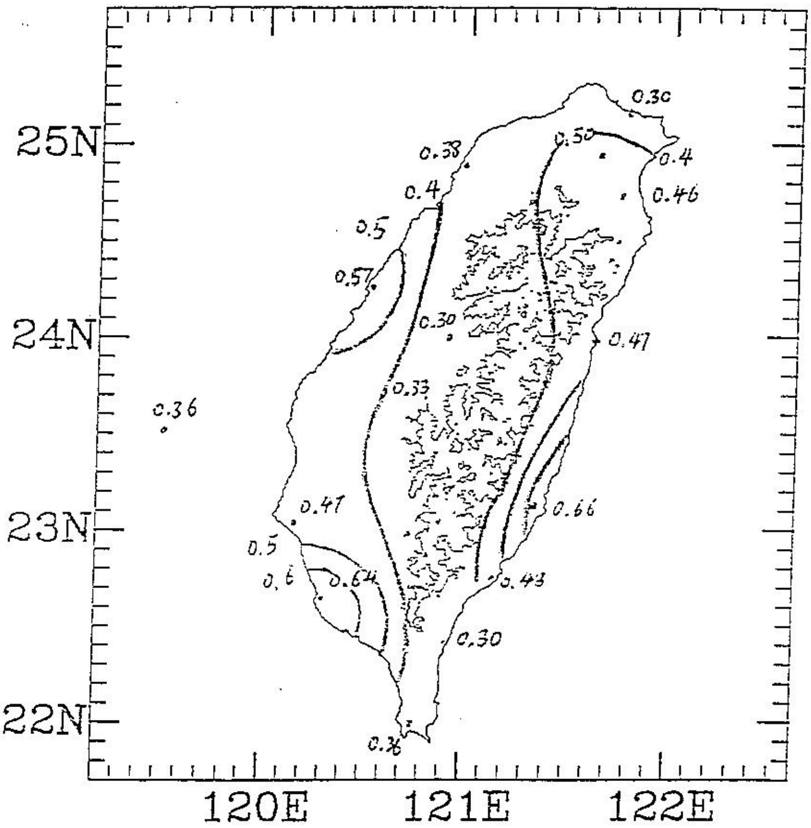


Fig. 4. Neural network cross-validation correlation skill for summer (JASO) rainfall at one-month lead.

that predicted by the CCA model, with a value hovering around 0.4. However, large differences between Figs. 3 and 4 are found in the western plain. No skill or even negative skill was found at Taichung, Tainan, and Kaohsiung in the CCA scheme but predictive skills have improved remarkably for those stations in the NN system (up to 0.64). Chengkung is another good example in which rainfall predictability increases substantially from 0.24 in Fig. 3 to 0.66 in Fig. 4. Similarly, there is also a notable improvement in skill for Taipei from 0.12 in the CCA scheme to 0.50 in the NN analysis. The above test clearly suggests that the NN model shows great promise as a new predictive tool in short-term climate prediction, particularly in the area where linear models have severe limitations in faithfully predicting real rainfall activity.

In the future, we intend to move into the next phase of the project, which is, designing a more efficient algorithm to achieve minimum error. In this regard, we will adopt the adaptive learning rate because it is faster and more reliable than the commonly used steepest descent method. This is to be followed by a proper weight initialization procedure which will get the learning process to start as fast as possible. Note that the fixed initial values for weights are used at present. With these additions, we will then test the model's performance in seasonal rainfall prediction experiments more rigorously.

References

Chu, P.-S., 1998: Short-term climate prediction of Mei-Yu rainfall for Taiwan using CCA. Int. J. Climatol., 18, 215-224.

Chu, P.-S., and W. Yan, 2000: Short-term climate predictability of summer rainfall for Taiwan. Proc. 2nd International Ocean and Atmosphere Conference. Taipei, Taiwan, 10-12 July, 53-58.

Hastenrath, S., L. Greischar, and J. V. Heerden, 1995: Prediction of the summer rainfall over

South Africa. J. Climate, 8, 1511-1518.

Navone, D., and H.A. Ceccatto, 1994: Predicting Indian monsoon rainfall-A neural network approach. Climate Dyn., 10, 305-312.

Silverman, D., and J.A. Dracup, 2000: Artificial neural networks and long-range precipitation prediction in California. J. Appl. Meteor., 39, 57-66.

Zorita, E., and H. Von Storch, 1999: The analog method as a simple statistical downscaling technique: Comparison with more complicated methods. J. Climate, 12, 2474-2489.