Manfred M. Fischer

Methodological Challenges in Neural Spatial Interaction Modelling: The issue of model selection

Paper

WSG Diskussionspapiere

WSG 65/99

Methodological Challenges in Neural Spatial Interaction Modelling:

The issue of model selection

Manfred M. Fischer

Institut für Wirtschafts- und Sozialgeographie

Wirtschaftsuniversität Wien

Department of Economic and Social Geography

Vienna University of Economics and Business Administration

WSG Discussion Papers

Abteilung für Theoretische und Angewandte Wirtschafts- und Sozialgeographie
Institut für Wirtschafts- und Sozialgeographie
Wirtschaftsuniversität Wien

Vorstand: o.Univ.-Prof. Dr. Manfred M. Fischer
A - 1090 Wien, Augasse 2-6, Tel. ++43-(0)1-31336-4836

Redaktion: Univ.-Ass. Dr. Petra Staufer

WSG 65/99

Methodological Challenges in Neural Spatial Interaction Modelling:

The issue of model selection

Manfred M. Fischer

WSG-Discussion Paper 65

February 1999

Gedruckt mit Unterstützung
des Bundesministerium
für Wissenschaft und Verkehr
in Wien

WSG Discussion Papers are interim
reports presenting work in progress
and papers which have been submitted
for publication elsewhere.

ISBN 3 85037 080 1

1. Introduction

Spatial interaction modelling is a well established field in geography and regional science. Since the pioneering work of Wilson (1970) on entropy maximization, however, there has been surprisingly little innovation in the design of spatial interaction models. The principal exceptions include the competing destinations version of Fotheringham (1983), the use of genetic algorithms to breed new forms of spatial interaction models, either directly (Openshaw 1988) or by genetic programming (Turton, Openshaw and Diplock 1997), and the design of neural spatial interaction interaction models (Fischer and Gopal, 1994, Gopal and Fischer 1993, Openshaw 1993). Neural spatial models are termed neural in the sense that they are based on neural computational approaches, inspired by neuroscience. They are more closely related to spatial interaction models of the gravity type, and under commonly met conditions they can be viewed as a special class of general feedforward neural network models with a single hidden layer and sigmoidal transfer functions (see Fischer 1998). Rigorous mathematical proofs for the universality of such feedforward neural network models (see, among others, Hornik, Stinchcombe and White 1989) establish the neural spatial interaction models as a powerful class of universal approximators for spatial interaction flow data.

Learning from examples, the problem for which neural networks were designed for to sove, is one of the most important research topics in artificial intelligence. A possible way to formalize learning from examples is to assume the existence of a function representing the set of examples and, thus, enabling to generalize. This can be called a function reconstruction from sparse data (or in mathematical terms, depending on the required precision, approximation or interpolation, respectively). Within this general framework, the central issues of interest are the representational power of a given network model (or, in other words, the problem of model selection) and the procedures for obtaining the optimal network parameters (see Fischer, Hlavackova-Schindler and Reismann 1998). No doubt, the tasks of parameter estimation and model selection are of crucial importance for the success of real world neural network applications. Model selection or the specification of a network topology is a key methodological issue and the primary focus in this contribution. Up to now, this issue has been highly neglected in spatial interaction modelling. Notable exceptions include Fischer and Gopal (1994) using cross-validation training and Fischer and Leung (1998) interweaving a genetic search for finding an optimal network topology with gradient-based backpropagation learning for determining the network parameters.

The contribution is organized as follows. First, a summarized description of single hidden layer neural spatial interaction is given in the next section. The goal of model selection is to optimize the complexity of the model in order to achieve the best generalisation. Considerable insight into this problem is provided in section 3 by introducing the concept of the bias-variance trade-off, in which the

generalization error is decomposed into the sum of the bias squared plus the variance. A neural spatial interaction model that is too simple will have a large bias and smooth out some of the underlying structure in the data [corresponding to high bias], while one model that has too much flexibility in relation to the particular data set will overfit the data and have a large variance. In either case, the performance of the model on new data will be poor. This highlights the need to optimize the complexity in the model selection process in order to achieve the best generalization of a model. Three principal ways to controlling the complexity of a model and, thus, to direct model search are discussed in section 4: *network pruning*, the use of *penalty terms*, and *stopped* or *cross-validation training*. The first two approaches can be viewed as variations of long established statistical techniques corresponding in the case of pruning to specification searches, and with respect to penalty terms as regularization or biased regression (Leamer 1979, Eubank 1988, Hanson and Pratt 1989, White 1989, Moody 1992). The procedure of cross-validation training seems to be one of the true innovations to come out out of neural network research. The model chosen does not require here the training process to converge, rather the training process is used to perform a directed search of parameter space to find a model with superior generalization performance.

2. Neural Spatial Interaction Models

Suppose we are interested in approximating a N-dimensional spatial interaction function $F:\Re^N \rightarrow \Re$, where $\Re^N$ as N-dimensional Euclidean real space is the input space and $\Re$ as 1-dimensional Euclidean real space is the output space. This function should estimate spatial interaction flows from regions of origin to regions of destination. The function F is not explicitly known, but given by a finite set of samples $S=\{(x^k, y^k), k=1, ..., K\}$ so that $F(x^k)=y^k$, k=1, ..., K. The set S is the set of pairs of input and output vectors. The task is to find a continuous function that approximates set S. In real world applications, K is a small number and the samples contain noise.

To approximate F, we consider the class of neural spatial interaction models Ω with one hidden layer, three input units, J hidden units and one output unit as suggested Fischer and Gopal (1994). The three input units correspond to the independent variables of the classical unconstrained spatial interaction model of the gravity type. They represent measures of origin propulsiveness, destination attractiveness and spatial separation. The output unit corresponds to the dependent variable of the classical model and represents the spatial interaction flows from origin to destination.

Ω consists of a composition of transfer functions so that the (single) output of Ω is

$$\Omega(\mathbf{x}, w) = \psi\left(\sum_{j=0}^{J} \alpha_j \phi_j\left(\sum_{n=0}^{3} \beta_{jn} x_n\right)\right) \qquad (1)$$

Vector $\mathbf{x}=(x_1, x_2, x_3)$ is the input vector argumented with a bias signal x_0 that can be thought as being generated by a 'dummy unit' whose output is clamped at 1. The β_{jn}'s represent input to hidden connection weights and the α_j's hidden to output weights (including the biases). The symbol w is a convenient shorthand notation of the d=(5J+1)-dimensional vector of all the α_j and β_{jn} network weights and biases (i.e. the model parameters). $\Phi_j(.)$ and $\psi(.)$ are differentiable non-linear transfer functions of the hidden units j=1, ..., J and the output unit, respectively.

The goal of learning is to find suitable values w^* for the network weights of the model such that the underlying mapping $F:\Re^3\rightarrow\Re$ represented by the training set $S=\{(x^k, y^k), k=1, ..., K\}$, is approximated or learned, where k is the index of the training instance. y^k, k=1, ..., K are scalars representing the desired network output (i.e. the spatial interaction flows) corresponding to x^k, k=1,..., K. The process of determining optimal parameter values is called training or learning and can be formulated in terms of minimization of an appropriate error function *E* to measure the degree of approximation with respect to the actual setting of network weights. The most common error function is the squared-error function of the patterns over the finite set of training data, so that the parameter estimation problem may be defined as the following minimization problem

$$\min_{\omega} E(\omega, S) = \min_{\omega} \sum_{(x^k, y^k)\in S} \left(\Omega(x^k, \omega) - y^k\right)^2 \qquad (2)$$

where the minimization parameter is the weight vector w defining the search space. In this way, the problem of network training has been formulated in terms of the minimization of the error function *E*. This error function is a function of the adaptive model parameters, i.e. network weights and biases. The derivatives of this function with respect to the model parameters can be obtained in a computationally efficient way using the propogation technique (see, e.g., Gopal and Fischer 1996; and Fischer and Staufer 1999, for more details on the equations of this technique).

The minimization of continuous differentiable functions of many variables is a problem that has been widely studied, and many of the non-linear minimization algorithms available are directly applicable to the training of neural spatial interaction models as defined by equation (1). The general scheme of these algorithms can be formulated as follows

(i) Choose an initial vector w in parameter space and set $\tau=1$,

(ii) Determine a search direction $\mathbf{d}(\tau)$ and a step size $\eta(\tau)$ so that

$$E(w(\tau) + \eta(\tau)\ d(\tau)) < E(w(\tau)) \qquad \tau = 1, 2, ...; \qquad (3)$$

(iii) Update the parameter vector

$$w(\tau+1) = w(\tau) + \eta(\tau)\ d(\tau) \qquad \tau = 1, 2, ...; \qquad (4)$$

(iv) If $\frac{dE(w)}{dw} \neq 0$ then set $\tau=\tau+1$ and go to (ii), else return $w(\tau+1)$ as the desired minimum.

It is important to stress that the objective of network training is not to learn an exact representation of the training data $(x^k, y^k) \in S$ itself, but rather to build an approximation of the process that generates the data. This is important if the model to exhibit good-out-of-sample (generalization) performance in view of novel data.

3. The Model Selection Problem

One of the major issues in neural spatial interaction modelling includes the problem of selecting an appropriate member of the model class Ω in view of a particular real world application. This model specification problem includes (i) the choice of appropriate transfer functions Φ_j and ψ, and (ii) the determination of an adequate network topology of Ω, i.e. the number, J, of hidden units.

Without loss of generality, let us assume the transfer functions $\Phi_j(.)=\Phi(.)=\psi(.)$ for all j=1, ..., J, and equal to the logistic function and, thus, consider the special class $\Omega_L(\mathbf{x}, w)$ of functions $\Omega(\mathbf{x}, w)$:

$$\Omega_L(\mathbf{x}, w)=\Big\{1+\exp[-\sum_{j=0}^{J} \alpha_j (1+\exp(-\sum_{n=0}^{3} \beta_{jn} x_n))^{-1}]\Big\}^{-1} \tag{5}$$

Then the only aspect of the model structure that remains to be determined is the number J of hidden units. Minimization of an error function such as (2) for determining values for the connection parameters and biases in a neural spatial interaction model is unable to determine the optimum size of J, because an increase in J - or, in other words, in the number of connection parameters - will generally allow a smaller value of the error to be found. The goal of model selection is to optimize the complexity of the model in order to achieve the best generalization.

Considerable insight into this phenomenon can be achieved by introducing the concept of the bias-variance trade-off, in which the generalization error is decomposed into the sum of the bias squared plus the variance. Following Bishop (1995) the sum-of-squares error, in the limit of an infinite data set S, can be written in the form of

$$E(w, S) = \frac{1}{2} \int (\Omega_L(\mathbf{x}, w)-\langle y|\mathbf{x}\rangle)^2 \, p(\mathbf{x}) \, d(\mathbf{x}) + \frac{1}{2} \int \big(\langle y^2|\mathbf{x}\rangle-\langle y|x\rangle^2\big) \, p(\mathbf{x}) \, d(\mathbf{x})) \tag{6}$$

in which $p(\mathbf{x})$ is the unconditional density of the input data, $\langle \mathbf{y}|\mathbf{x}\rangle$ denotes the conditional average, or regression, of the target data $\mathbf{y}$ given by

$$\langle \mathbf{y}|\mathbf{x}\rangle \equiv \int \mathbf{y} \, p(\mathbf{y}|\mathbf{x}) \, d\mathbf{y} \tag{7}$$

where $p(\mathbf{y}|\mathbf{x})$ is the conditional density of the target variable $\mathbf{y}$ conditioned on the output vector $\mathbf{x}$ similarly

$$\langle \mathbf{y}^2|\mathbf{x}\rangle \equiv \int \mathbf{y}^2\, p(\mathbf{y}|\mathbf{x})\, d\mathbf{y} \tag{8}$$

Note that the second term in (6) is independent of the spatial interaction function $\Omega_L(\mathbf{x}, w)$ and, thus, is independent of the network weights w. The optimal model, in the sense of minimizing the sum-of-squares error is the one that makes the first term in (6) vanish, and is given by $\Omega_L(\mathbf{x}, w) = \langle \mathbf{y}|\mathbf{x}\rangle$. The second term represents the intrinsic noise in the data and sets a lower limit on the error that can be achieved.

In real world application contexts we have to deal with the problems arising from a finite size data set. Suppose, we consider a training set S consisting of K patterns that we utilize to determine the neural spatial interaction function $\Omega_L(\mathbf{x}, w)$. Now consider a whole ensemble of possible data sets, each containing K patterns, and each taken from the same fixed joint distribution $p(\mathbf{x}, \mathbf{y})$. The optimal network model is given by the conditional average $\langle \mathbf{y}|\mathbf{x}\rangle$. A measure of how close the actual spatial interaction function $\Omega_L(\mathbf{x}, w)$ is to the desired one is given by the integrand of the first term in (6): $(\Omega_L(\mathbf{x}, w)-\langle \mathbf{y}|\mathbf{x}\rangle)^2$. The value of this quantity will depend on the particular data set S on which it is trained. We can eliminate this dependence by considering an average over the complete ensemble of data sets, that is

$$\mathcal{E}_S\,[(\Omega_L(\mathbf{x}, w)-\langle \mathbf{y}|\mathbf{x}\rangle)^2] \tag{9}$$

where $\mathcal{E}_S(.)$ denotes the expectation [ensemble average]. If model Ω_L was always a perfect predictor of the regression function $\langle \mathbf{y}|\mathbf{x}\rangle$ then this error would be zero. A non-zero error can arise essentially due to two distinct reasons: First, it may be that the model Ω_L is on average different from the regression function. This is termed *bias*. Second, it may be that Ω_L is very sensitive to the particular data set S, so that, a given $\mathbf{x}$, it is larger than the required value for some data sets, and smaller for other data sets. This is called *variance*. We can make the decomposition into bias and variance explicit by writing (9) in a somewhat different, but mathematically equivalent form (see Bishop 1995):

$$\mathcal{E}_S\,[(\Omega_L(\mathbf{x}, w)-\langle y|\mathbf{x}\rangle)^2] = \mathcal{E}_S\,(\Omega_L(\mathbf{x}, w)-\langle y|\mathbf{x}\rangle)^2 + \mathcal{E}_S\,(\Omega_L(\mathbf{x}, w)-\ \mathcal{E}_S\,(\Omega_L(\mathbf{x}, w)))^2 \tag{10}$$

where the first term of the right hand side of the equation denotes the bias squared and the second term the variance. The bias measures the extent to which the average over all data sets of the spatial interaction function differs from the desired function $\langle \mathbf{y}|\mathbf{x}\rangle$. Conversely, the variance measures the extent to which Ω_L is sensitive to the particular choice of data sets.

A neural spatial interaction model that is too simple [i.e. small J], or too inflexible, will have a large bias and smooth out some of the underlying structure in the data [corresponding to high bias], while one that has too much flexibility in relation to the particular data set will overfit the data [corresponding to high variance] and have a large variance. In either case, the performance of the network on new data [i.e. generalization performance] will be poor. This highlights the need to optimize the complexity in the model selection process in order to achieve the best generalization.

4. Model Selection Techniques

Both the theoretical and practical side of the model selection problem has been intensively studied in the field of neural networks and a vast array of methods have been suggested to perform this task. Most approaches view model selection as a process consisting of a series of steps that are performed independently.

Step 1: The first step consists of choosing a specific parametric representation that is oversized in comparison to the size of the training set used.

Step 2: Then in the second step either an error function such as *E* [possibly including a regularization term] is chosen directly, or in a Bayesian setting, prior distributions on the elements of the data generation process (noise, model parameter, regularizers, etc.) are specified from which an objective function is derived.

Step 3: Utilizing the error function specified in *Step 2*, the training process is started and continued until a convergence criterion is fulfilled. The resulting parametrization of the given model architecture is then placed in a pool of model candidates from which the final model will be chosen.

Step 4: To avoid overfitting, model complexity must be limited. Thus, the next step usually consists of modifying the network model architecture [for example, by pruning weights], or of the penalty term [for example, by changing its weighting in the objective function], or of the Bayesian prior distributions. The last two modifications then lead to a modification of the objective function. It is worthwhile noting that this establishes a new framework for the training process that is then restarted and continued until convergence, yielding another model for the pool.

This process is iterated until the model builder is satisfied that the pool contains a reasonable diversity of model candidates, that are then compared with each other using some estimator of generalization ability, for example, the average relative variance [i.e. a normalized mean squared error metric] given by (Fischer and Gopal 1994)

$$\mathrm{ARV}\,(S) = \frac{\sum_{(x^k,y^k)\in S} \left(y^k - \Omega_L\left(x^k, w\right)\right)^2}{\sum_{(x^k,y^k)\in S} \left(y^k - \bar{y}\right)^2} \tag{11}$$

where **y** denotes the target vector and $\bar{y}$ the average over K desired values in S.

The methods employed for training may be very sophisticated (see, for example, Fischer and Staufer 1999). In contrast to this, the choice and modification of the network model architecture and

objective function is generally ad hoc, or is directed by a search heuristic (see, for example, Openshaw 1993). In this contribution three principal approaches for directing model modification and selection are considered.

The first approach is by use of *pruning techniques*. The principal idea of pruning is to reduce the number of model parameters by removing dispensable ones. Thus, pruning techniques function by training an oversized neural network model with a fixed, but larger J to a minimum of $E(\mathrm{w}, \mathrm{S})$, then testing elements of the model, such as connection parameters, for relevance. Those elements with poor test results are then deleted and the modified network model is retrained. In this approach one uses the information in an existing model to direct the search to the best 'neighbouring' model.

Clearly, various choices have to be made utilizing this approach. The most important is how to decide which parameter weights should be removed. The decision is generally based on some measure of the relative importance, or saliency, of different weight parameters. The simplest concept of saliency is to suppose that small weights are less important than large weights, and to use the absolute magnitude of a parameter value as a measure of its importance, under the assumption that the training process naturally forces non-relevant weights into a region around zero.

A major shortcoming of this pruning technique is its weak theoretical motivation. Since parameter estimation is defined in terms of the minimization of the error function *E*, it is natural to use the same error function to find a more principled definition of saliency. Especially, we could define the saliency of a model parameter as the change in the error function that results from deletion of that parameter. This could be implemented by direct evaluation so that, for each parameter in the trained network model in turn, the parameter is temporarily set to zero and the error function re-evaluated. Though conceptually attractive, such an approach will be computationally demanding in the case of larger neural spatial interaction models.

Consider instead the change in the error function due to small changes in the parameter values (Le Cun, Denker and Solla 1990). If the paremeter w_i is changed to $w_i+\delta w_i$, then the corresponding change in the error function *E* is given by

$$\delta E(\mathrm{w}) = \sum_{i=1}^{5J+1} \frac{\partial E(\mathrm{w})}{\partial \omega_i} \delta\omega_i + \frac{1}{2} \sum_{i=1}^{5J+1} \sum_{j=1}^{5J+1} \mathrm{H}_{ij} \delta\omega_i \delta\omega_j + O(\delta\omega^3) \qquad (12)$$

where H_{ij} denote the elements of the Hessian matrix

$$\mathrm{H}_{ij} = \frac{\partial^2 E(\mathrm{w})}{\partial\omega_i \partial\omega_j} = \nabla^2 E(\mathrm{w}). \qquad (13)$$

If we assume that the parameter estimation process has converged, then the first term in (12) will vanish. Le Cun, Denker and Solla (1990) approximate the Hessian by discarding the non-diagonal terms. Neglecting the higher order terms in the expansion then (12) reduces to the form

$$\delta E(w) = \frac{1}{2} \sum_{i=1}^{5J+1} H_{ij} \delta \omega_i^2 . \qquad (14)$$

If a parameter having an initial value w_i is set to zero, then the increase in E will be approximately given by the quantities $\frac{1}{2} H_{ij} \omega_i^2$ that can be interpreted as a statistical significance measure (see Finoff, Hergert and Zimmerman 1993).

An implementation of this pruning procedure would characteristically consists of the following steps:

- *first*, select a relatively large initial network model [i.e. relatively large J],
- *second*, train the network model in the usual way [for example by backpropagation of gradient descent errors] until some stopping criterion is satisfied,
- *third*, compute the second derivatives H_{ij} for each of the parameters, and thus evaluate the saliences $\frac{1}{2} H_{ij} \omega_i^2$,
- *fourth*, sort the parameters by saliency and delete some of the low-saliency weights,
- *fifth*, go to the second step and repeat until some overall stopping criterion is satisfied.

Clearly, there are various choices to be made. The most important consideration, however, is to decide upon an appropriate number of parameters to be removed. The choice can be influenced by the number of pruning steps already performed as well as by visual inspection of the distribution of the test measure (see Finoff, Hergert and Zimmmerman 1993). If this problem is solved satisfactorily the pruning technique that is generally performed interactively reduces overfitting and improves generalization of neural spatial interaction models.

The second approach for directing network architecture modification and selection is through the use of *regularization* which involves the addition of an extra term $R(w)$ to the error function $E(w)$ which is designed to penalize mappings that are not smooth. With a sum-of-squares error function, the total error function, $E(w)$, to be minimized becomes

$$\tilde{E}(\omega) = \sum_{(x^k, y^k) \in S} (\Omega(x^k, w) - y^k)^2 + \mu R(w) \qquad (15)$$

The parameter μ [$\mu \in [0, \infty)$] controls the degree of regularization, i.e. the extent to which the penalty term $R(w)$ influences the form of the solution. Training is performed by minimizing the total error function $\tilde{E}(\omega)$ that requires the derivatives of $R(w)$ with respect to the model parameters to be computed efficiently. A spatial interaction function Ω_L that provides a good fit to the training data will give a small value for $\tilde{E}(w)$, while one that is very smooth will give a small value for $R(w)$. The resulting network model is a compromise between fitting the data and minimizing $R(w)$. One of the simplest regularizers $R(w)$ is called *weight decay* and consists of the sum of squares of the adaptive model parameters

$$R(w) = \frac{1}{2} \sum_{i=1}^{5J+1} \omega_i^2 \tag{16}$$

the first derivative of which leads to the weight decay in the weight updates (see Hanson and Pratt 1989). The use of this form of regularizer corresponds to ridge regression in convertional curve fitting. Hinton (1987) has emprically shown that such a regularizer can lead to significant improvements in network generalization.

One of the difficulties of the simple regularizer (17) is that it tends to favour many small parameter values rather than a few large ones. This problem can be overcome by using a modified penalty term of the form

$$R(w) = \sum_{i=1}^{5I+1} \frac{\omega_i^2}{\hat{\omega}^2 + \omega_i^2} \,. \tag{17}$$

where $\hat{\omega}$ corresponds to a parameter that sets a scale usually chosen by hand to be of order unity. Use of this penalty term has been called *weight elimination* (Weigend, Huberman and Rumelhart 1990). It will tend to favour a few large parameter values rather than many small ones, and, thus, is more likely to eliminate parameters from the model than (16). This leads to a form of network model pruning which is combined with the training process itself rather than alternating with it as in the case of pruning techniques.

A principal alternative to regularization and weight pruning as a way of controlling the effective complexity of a neural spatial interaction model is the procedure of *stopped* or *cross-validation* training. This method, in which an oversized network model is trained until the error on a further validation data set deteriorates, then training is stopped, is a true innovation coming out of neural network research since model selection does not require convergence of the training process. The training process is used here to perform a directed search of the parameter space for a model that does not overfit the data and, thus, demonstrates superior generalization performance. Various theoretical and empirical results have provided strong evidence for efficiency of cross-validation training (Weigend, Rumelhart and Huberman 1991, Baldi and Chauvin 1991, Finnoff 1991, Fischer and Gopal 1994). Although many questions remain, a picture is starting to emerge as to the mechanisms that are

responsible for the effectiveness of this procedure. In particular, it has been shown that stopped training has the same sort of regularization effect (i.e. reducing model variance at the cost of bias) that penalty terms provide.

5. Summary and Outlook

Building a neural spatial interaction model involves two distinct tasks: model choice [determination of a network structure] and parameter estimation. The specification of an appropriate network topology is a key issue because it governs the capability of the network model to provide an adequate approximation of the input-output relationships. This contribution provided insights into why the complexity of the model has to be optimized in order to achieve the best generalization. In the case of neural spatial interaction models, the complexity can be varied by changing the number of adaptive parameters. This is called structure stabilization.

Three major approaches to controlling the complexity of a neural spatial interaction model have been considered: pruning, regularization and cross-validation. As noted above, the primary innovation in the use of cross-validation training to perform the task of model selection is to consider any parametrization of a given network architecture as a potential model. Evidently, an exhaustive search of all parametrisations of a given network architecture will be just as inefficient as an exhaustive search over a large class of potential network architectures (prior distributions etc.). In the case of cross-validation training, the training process is utilized to direct the search for potential models in parameter space. This can be seen as a variant of network pruning for model selection that functions by training a network model with a fixed model architecture to a minimum, then testing model parameters for relevance. Those parameters with poor test results are then removed and the modified model is retrained.

Since the basic effect of regularization is also to reduce model complexity and consequent model varaince, one sees that there is a close relationship in the instrumental effects of cross-validation training, pruning and regularization. The question remains whether - or under what circumstances - any one of these principal approaches to model selection generates superior results.

Acknowledgement: The author gratefully acknowledges the grant no. P12681-INF provided by the Fonds zur Förderung der Wissenschaftlichen Forschung (FWF).

References

Baldi, P. and Chauvin, Y. (1991): Temporal evolution of generalization during learning in linear networks, *Neural Computation* 3, 589-603.

Bishop, C.M. (1995): *Neural Networks for Pattern Recognition*, Oxford, Clarendon Press.

Eubank, R. (1988): *Spline Smoothing and Nonparametric Regression*. New York, Marcel Dekker.

Finnoff, W. (1991): Complexity measures for classes of neural networks with variable weight bounds, *Proceedings of the International Joint Conference on Neural Networks, IJCNN'91*, 2624-2630, Singapore, IEEE.

Finnoff, W., Hergert, F. and Zimmermann, H.-G. (1993): Improving model selection by nonconvergent methods, *Neural Networks* 6, 771-783.

Fischer, M.M. (1998): Computational neural networks - a new paradigm for spatial analysis, *Environment and Planning A* 30(10), 1873-1892.

Fischer, M.M. and Gopal, S. (1994): Artificial neural networks. A new approach to modelling interregional telecommunication flows, *Journal of Regional Science* 34 (4), 503-527.

Fischer, M.M. and Leung, Y. (1998): A genetic-algorithms based evolutionary computational neural network for modelling spatial interaction data, *Annals of Regional Science* 32(3), 437-458.

Fischer, M.M. and Staufer, P. (1999): Optimization in an error backpropagation neural network environment with a performance test on a spectral pattern classification problem, *Geographical Analysis* 31, in press.

Fischer, M.M., Hlavácková-Schindler, K. and Reismann, M. (1998): A global search procedure for parameter estimation in neural spatial interaction modelling, submitted for publication to *The Papers in Regional Science.*

Fotheringham, A.S. (1983): A new set of spatial interaction models: The theory of competing destinations, *Environment and Planning A* 15, 15-36.

Gopal, S. and Fischer, M.M. (1993): Neural net based interregional telephone traffic models, *Proceedings of the International Joint Conference on Neural Networks, IJCNN'93*, 2041-2044, Nagoya, IEEE.

Gopal, S. and Fischer, M.M. (1996): Learning in single hidden-layer feedforward network models, *Geographical Analysis* 28 (1), 38-55.

Hanson, S.J. and Pratt, L.J. (1989): Comparing biases for minimal network construction with back-propagation. In Touretzky, D.S. (ed.), *Advances in Neural Information Processing*, 177-185. San Mateo, Morgan Kaufmann.

Hinton, G.E. (1987): Learning translation invariant recognition in massively parallel networks. In Bakker, J.W. de, Nijman, A.J. and Treleaven, P.C. (eds.) *Proceedings PARLE Conference on Parallel Architectures and Languages Europe*, 1-13. Berlin, Springer.

Hornik, K., Stinchcombe, M. and White, H. (1989): Multilayer feedforward networks are universal approximators, *Neural Networks* 2, 359-366.

Le Cun, Y., Denker, J.S. and Solla, S.A. (1990): Optimal brain damage. In Touretzky, D.S. (ed.), *Advances in Neural Information Processing*, 598-605. San Mateo, Morgan Kaufmann.

Learner, E.E (1979): *Specification Searches*. New York, Wiley.

Moody, J. (1992): Generalization, weight decay and architecture selection for nonlinear learning systems. In Moody, J., Hanson, J. and Lippmann, R. (eds.) *Advances in Neural Information Processing Systems*, 471-479. San Mateo, Morgan Kaufman.

Openshaw, S. (1988): Building an automated modelling system to explore a universe of spatial interaction models, *Geographical Analysis* 20(1), 31-46.

Openshaw, S. (1993): Modelling spatial interaction using a neural net. In Fischer, M.M., Nijkamp, P. (eds.) *Geographic Information Systems, Spatial Modelling, and Policy Evaluation*, 147-164. Berlin, Springer.

Turton, I., Openshaw, S. and Diplock, G. (1997): A genetic programming approach to building new models relevant to GIS. In Kemp, Z. (ed.), *Innovations in GIS*, 89-102. London, Taylor and Francis.

Weigend, A., Huberman, B. and Rumelhart, D.E. (1990): Predicting the future: A connectionist approach, *International Journal of Neural Systems* 1, 193-209.

Weigend, A., Rumelhart, D.E. and Huberman, B. (1991): Generalization by weight elimination with application to forecasting. In Lippman, R., Moody, J. and Touretzky, D. (eds.) *Advances in Neural Information Processing Systems*, 875-882. San Mateo, Morgan Kaufmann.

White, H. (1989): Learning in artificial neural networks: A statistical perspective, *Neural Computation* 1, 425-464.

Wilson, A.G. (1970): *Entropy in Urban and Regional Modelling*. London, Pion.