NEURAL NETWORKS FOR CROSS-SECTIONAL EMPLOYMENT FORECASTS: A COMPARISON OF MODEL SPECIFICATIONS FOR GERMANY

Roberto Patuelli,[1] Aura Reggiani,[2] Peter Nijkamp[3] and Norbert Schanne[4]

[1] Institute for Economic Research (IRE), University of Lugano, Switzerland;
The Rimini Centre for Economic Analysis, Italy
[2] Department of Economics, University of Bologna, Italy
[3] Department of Spatial Economics, VU University Amsterdam, The Netherlands
[4] Institute for Employment Research (IAB), Nuremberg, Germany

Abstract

In this paper, we present a review of various computational experiments – and consequent results – concerning Neural Network (NN) models developed for regional employment forecasting. NNs are widely used in several fields because of their flexible specification structure. Their utilization in studying/predicting economic variables, such as employment or migration, is justified by the ability of NNs of learning from data, in other words, of finding functional relationships – by means of data – among the economic variables under analysis.

A series of NN experiments is presented in the paper. Using two data sets on German NUTS 3 districts (326 and 113 labour market districts in the former West and East Germany, respectively), the results emerging from the implementation of various NN models – in order to forecast variations in full-time employment – are provided and discussed. In our approach, single forecasts are computed by the models for each distinct district. Different specifications of the NN models are first tested in terms of: (a) explanatory variables; and (b) NN structures. The average statistical results of simulated out-of-sample forecasts on different periods are summarized and commented on.

In addition to variable and structure specification, the choice of NN learning parameters and internal functions is also critical to the success of NNs. Comprehensive testing of these parameters is, however, limited in the literature. A sensitivity analysis is therefore carried out and discussed, in order to evaluate different combinations of NN parameters. The paper concludes with methodological and empirical remarks, as well as with suggestions for future research.

Pn322rparns

1 Introduction

Forecasting in economics has been on a rising edge over the years, because of the increased need, in particular by policy-making agencies, for optimal policy intervention and stimuli. In particular, because of the ongoing shift towards tailor-made region-specific policies, meso-economic (sectoral or regional) forecasts are in great demand. On the other hand, new problems tend to arise in conjunction with new forecasting tasks, such as: (a) the imbalance between the increased number of regions to forecast for and the time span of the observations available; and (b) the complex dynamics and economic interdependencies influencing economic performance, which are often difficult to measure and which create difficult specification issues in inferential statistics.

A non-conventional and increasingly popular approach to economic forecasting that may overcome some of the above problems is offered by the family of mathematical methods of 'neural networks' (NNs). NNs are optimization algorithms, which have the capacity to learn functional relationships from the data and replicate them for out-of-sample forecasting. This characteristic makes them a flexible statistical tool for the solution of complex socio-economic problems. Labour market developments are a good example of such complex forecasting issues, as there are many forces at work (demand-supply, sectoral, geographic, institutional) which may lead to complex evolutionary patterns that cannot be handled by standard linear modelling approaches. In addition to having a non-linear nature, NNs do not require *a priori* modelling hypotheses, which are sometimes difficult to formulate, in particular when the implications of the variables concerned are not fully known, or when insufficient insight into the forces at work exists.

While NNs have several advantages, they also have drawbacks, such as the limited behavioural-theoretical interpretation of their results. However, interpretation issues are, in our case, less opportune, because our focus is on forecasting. Another caveat in the use of NNs is that they have been shown to be sensitive to the choice of the parameters implemented within the algorithms used (see, for example, Hagan et al. 1996).

The non-explicit behavioural foundation of the NN models in economic theory – which precludes a straightforward theoretically-based specification analysis of models – leads to the need to explore different – sometimes complementary – model specifications in an NN context so as to test the robustness of forecasting results. The objective of the present paper is to investigate the sensitivity of NN models – developed for regional employment forecasts – to different model specifications and to changing parameter values, with the final aim being to maximize the forecasting potential of the models under consideration. We use German regional employment variables as a case study, and develop and estimate a set of NN models by utilizing different inputs.

The paper is structured as follows: Section 2 provides a very brief pedagogical description of the working of NNs. Section 3 illustrates the implementation of a set of NNs models developed for regional employment forecasting. Section 4 presents a sensitivity analysis, which was carried out in order to test different combinations of learning parameters and internal functional forms. Section 5 reviews, on the basis of the sensitivity analysis findings, a set of NN models, and presents an evaluation and comparative discussion of their statistical performance. Finally, Section 6 draws methodological and empirical conclusions, as well as suggestions for future research.

2 Neural Networks

NNs are often referred to as a 'black box' approach. Though they are regarded as such in particular in social sciences, because of their no-theory modelling characteristics, NNs are not an obscure tool. The internal functions that process the information inputs, as well as the algorithms that determine the direction and the degree of interaction of the factors, can be clearly explained formally and mathematically. On top of it, they can be proven to be consistent with standard goodness-of-fit conditions (see, for example, Schintler and Olurotimi 1998).

NNs (Rosenblatt 1958; Werbos 1974) are optimization tools that – originally – aimed to replicate the simultaneous information processing and data-driven learning seen in biological networks. A generic NN can be defined as a multilevel system of computation units (or neurons), which are distributed in interlinked layers. The computation units can either refer to the input variables (which are contained in the first layer) or to the output variables (in the last layer), or be used for intermediate calculation (if present, in the hidden layers). A NN with no hidden layers is called a 1-layer structure, as the output layer is usually not counted, since it does not take part in the data computation. Accordingly, a NN with one hidden layer has a 2-layer structure, and so on. In feedforward NNs, every unit is connected to all units in the successive layer, and connections only go in one direction – forward (other types of NNs, such as recurrent NNs, are not considered here).

Without loss of generality, in the univariate case, the output of the generic processing unit $u_{i,n}$ is obtained as follows:

$$u_{i,n} = \varphi\left(\mathbf{u}_{n-1}\right) = \Im\left(f\left(\mathbf{u}_{n-1}\right)\right), \tag{1}$$

where $\mathbf{u}_{n-1} = \left\{u_{1,n-1},...,u_{k,n-1}\right\}$ is the preceding layer of units, and the transfer function φ can be decomposed into two separate functions: the activation function $\Im$, and the integrator function f. The former computes the units' output, and is usually a (logistic) sigmoid (see Subsection 4.3) while the latter aggregates the information processed by the units of the preceding layer (in Equation (1), $\mathbf{u}_{n-1}$) connected to unit $\mathbf{u}_n$. This is often done by means of a weighed sum of the type $v_{i,n} = f(\mathbf{u}_{n-1}) = \sum_j w_{ij,n-1} u_{j,n-1}$. The weights $w_{ij,n-1}$ used in the integrator function are recursively computed during the 'training' of the NN, guaranteeing the 'learning' process. They have an essential role, as the 'knowledge' generated by the NN is contained in the set of weights computed. Clearly, these weights have to be computed.

The backpropagation algorithm (BPA) (Rumelhart and McClelland 1986) is the algorithm most commonly used for the computation of the above weights. Learning from examples of inputs and outputs provided by the analyst, the NN identifies the relationships underlying the data. Such a class of NN models is usually called supervised NNs (unsupervised NNs are not discussed here). The learning process of the NN is given by the comparison between the output generated from Equation (1) in the output layer and the correct output. The obtained error is propagated backward through the network until the input layer, and the process is repeated, with consequent re-adjustments of the weights,[1] until a stopping condition is satisfied (for more details on the BPA, we refer to Rumelhart and McClelland 1986).

Although the process described does not require actions from the analyst, NNs are not completely autonomous. BPA networks tend to fall into local minima or to overfit the data (Zhang et al. 1998). Overfitting can happen when an excessive number of iterations is carried out, a situation that can be detected by observing a deterioration in the statistical error of the NN. A number of techniques can then be used to deal with this potential drawback, the most common being early stopping. In early stopping, the training of the network is stopped once the statistical error computed at each iteration reaches a slow convergence or increases. NNs were also shown to be sensitive to changes in the values of the learning parameters internal to the BPA, as well as to the activation function used (Klimasauskas 1991; Hagan et al. 1996). These aspects are discussed in Section 4. First, Section 3 will describe the implementation of NN models for regional employment forecasting, and the statistical results obtained for different model specifications.

[1] The starting set of weights is usually randomly defined, so to generate a large error in the first iteration and facilitate the convergence of the algorithm (Cooper 1999).

3 Neural Networks for Forecasting Regional Employment

The basic working of general NNs described in the preceding section has now to be fine-tuned to the use for regional employment forecasting. The variable we want to predict is the growth rate of fulltime employment in 439 NUTS-3 districts in Germany. We focus on forecasting biannual growth rates, that is, 2 years ahead $(t, t+2)$. We use panel data for the periods 1987–2004 and 1993–2004, for West and East Germany, respectively.[2] The nature of the data used is indeed the most important aspect of our experiments. Differently from conventional panel models (see, for example, Baltagi 2001), a standard NN does not include temporal correlation. Still, identifying time information in the models is critical in order to recognize time-specific shocks and, in the case of Germany, the continuing effects of the reunification. Therefore, the main problem faced in developing our models is: How can NNs recognize and treat the time correlation in the data? We choose to introduce time in our models by employing a time variable that identifies – by means of a text (string) variable – the years concerned. This approach is made possible by rescaling the text variable, that is, each year is associated with a numerical value within the (0,1) interval, therefore identifying a year specific intercept. An alternative but similar approach, based on the use of yearly dummy variables, was proven less preferable (see Patuelli et al. 2006; 2007). In addition to time, the main covariates employed in all models proposed later in the paper are the growth rates observed in fulltime employment, for the period $(t-2, t)$. We subdivide the employees in nine sectors, ranging from primary goods to services.

Starting from this baseline model (hereforth, Model B), in Section 5, five additional models are tested, which are obtained when more covariates are considered:

- Model BD uses a 9-point index of the level of urbanization and agglomeration of the districts (see Böltgen and Irmen 1997). This index aims to account for the different economic trends of urbanized, agglomerated and rural areas;
- Model BW has, as an additional variable, information on average regional daily wages of fulltime workers. The wage variable aims to capture the well known relationship between labour supply/demand and wages;
- Model BSS uses the competitive effect components computed by means of shift-share analysis (SSA) (Dunn 1960) for the nine economic sectors concerned. These components express the competitiveness – in terms of employment growth rates – of each region in each sector, compared with sectoral trends at the national level;
- Model BSSN uses competitive effect components, similarly to Model BSS, but computed according to the spatial shift-share approach, as described in Nazara and Hewings (2004). In spatial shift-share, the employment performance of regions is not compared to national performance, but to the one of neighbours,[3] so to capture spatial/economic correlation;
- Model BSSR uses modified competitive effects. These effects were computed by multiplying the components used in Model BSS by the respective regression coefficients obtained by means of (simplified) shift-share regressions carried out, for each year of data, as in Patuelli et al. (2006). The new effects ought to be a fine-tuning of the ones used in Model BSS.

[2] The data (on fulltime employment and average daily wages) used in our experiments have been provided by the German Institute for Employment Research (*Institut für Arbeitsmarkt und Berufsforfschung*, IAB). As these data are directly collected at single-firm level, they are expected to have low and non-systematic measurement errors. The employment data refer, for each year, to the second quarter (full quarterly data were not available for the experiments).

[3] In the spatial shift-share approach, we define as neighbours the three districts that provide the most commuters to each district. Neighbours beyond the former boundary between West and East Germany are not considered. The commuting data have been provided by Franz-Josef Bade (University of Dortmund, Germany), whom the authors greatly acknowledge.

The above models are estimated separately, for both West and East Germany, because of the different time span of the data (1987–2004 and 1993–2004, respectively). Several structures are tested for each model: (a) a 1-layer structure; (b) 2-layer structures with 5, 10, and 15 hidden units; and (c) a 3-layer structure with 5 hidden units in both hidden layers.[4] All NN models are validated on the years 1999 and 2000 for West Germany, and on the year 2000 for East Germany (because of the shorter data span). One of the above structures is chosen for each model, according to mean squared error (MSE) and mean absolute error (MAPE) values. Overfitting is avoided by means of early stopping of the training.

Once validated, the NN models are applied to more recent data (from 1999–2001 to 2002–04), obtaining, for each 2-year period, simulated out-of-sample forecasts. In Table 1, we present the average (pooled) statistical error found for the four periods.

The choice of the covariates to use is not the only relevant part of the process of developing an NN model. Because of the local minima search characteristic, NNs are known to have volatile performance, and a tendency to overfitting. The next section discusses the selection of appropriate NN parameters, by means of a sensitivity analysis.

4 Sensitivity Analysis

4.1 Preface

This section is concerned with describing – and testing – the main parameters and functions that are used internally to NNs. It is relevant to deal with concepts such as learning rate or activation function, since they greatly influence the performance of NNs models (see, for example, Hagan et al. 1996). In our case, the objective is to find the optimal combination of parameters in order to increase the forecasting potential of our models.

Sensitivity analyses of NN learning parameters or activation functions have been previously carried out (see, for example, in the case of neural spatial interaction models, Gopal and Fischer 1996). Srinivasan et al. (1994) experimented with different activation functions (symmetrical and non-simmetrical) and learning parameters, in the context of electrical load forecasting. However, no detailed results are presented emerging from their analysis. Gorr et al. (1994) used a grid search procedure for choosing learning rate values (jointly to the number of iterations), but did not test the suitability of alternative activation functions, as well as Sharda and Patil (1992). Generally, more attention is focussed on the choice of NN learning parameters, rather than on the choice of the activation function.

The sensitivity analysis illustrated in the following sections aims to evaluate the use of both different combinations of learning parameters (Section 4.2), and of varying activation functions (Section 4.3), so to provide a more complete overview of NN setting issues.

[4] There is no agreement in the literature on how to select the number of hidden units contained in the hidden layers. Tang and Fishwick (1993) suggest that the number of hidden units in a NN has an effect on its forecasting performance, but this effect does not seem to be significant (Zhang et al. 1998). Others suggest that a number of hidden units equal to the number of input units (in a 2-layer framework) would provide improved results (Chakraborty et al. 1992; Sharda and Patil 1992; Tang and Fishwick 1993). It is generally recommended to experiment, for each empirical application, with different NN configurations – proceeding 'at jumps' – so as to find heuristically the NN that fits best one's needs. This approach was followed in our experiments.

4.2 Learning Rate and Momentum

4.2.1 Description

The backpropagation algorithm (BPA) (see Section 2) can be seen as a gradient steepest descent method, an optimization method based on the search for local minima of functions (Zhang et al. 1998; see also Weisstein 2006). In order to use a gradient descent algorithm, a step size – that is, a scaling parameter – is necessary. In NNs, this is called 'learning rate' (LR), which, jointly with the momentum parameter, is crucial in determining the NN learning curve, in terms of potential, stability and computing time. Different combinations of the values given to the two parameters can generate significantly different results. Simply said, a NN's LR determines the magnitude of the correction that is applied, during the learning phase, when adjusting the weights of the computation units. On the other side, the momentum defines how lasting the corrections applied will be, that is, for how many iterations they will survive.

Learning rates assume positive values, which range from 0 to 1. On the one hand, large values imply a quick learning of the network. On the other hand, values that are too large may cause the NN to be unstable, therefore nullifying the learning carried out at previous iterations. Generally, unstable behaviour can be avoided for LR values smaller than 0.25. The drawback of using such small LR values is the longer computing time required for training.

The tricky nature of the LR parameter calls for empirical testing. In fact, the BPA is known to suffer from slow convergence, inefficiency and lack of robustness (Zhang et al. 1998). Furthermore, it can be very sensitive to the choice of the LR. Ideally, one should experiment with different values of LR, in order to find the most suitable one for the data at hand. Gorr et al. (1994) propose to use a search grid in order to test different LR values. Although more automated optimization procedures can be used in this regard (we refer, for example, for the discussion of adaptive LR to Section 4.2.3), a more conservative approach may be to manually adjust the LR values, starting from low values, which can be increased if the learning process is low.

The performance of the BPA can be improved by including an additional parameter, viz. momentum. The momentum parameter determines the lifespan of the corrections made to the NN weights during the training process. Its aim is to allow for greater values of the LR, therefore fastening convergence, while reducing the fluctuations of the BPA. The momentum parameter assumes values greater than (or equal to) 0, but smaller than 1.[5] On the one hand, momentum values that are close to 1 will increase the influence that previous corrections to the weights have on the current corrections. On the other hand, a NN with a momentum close to 0 will mainly (or 'only', in the case of 0) rely on the current corrective term, at each stage of the training. For example, a momentum value set at 0.5 means that 50 per cent of the weight adjustment, at each stage, will be on the basis of the current error, while the remaining 50 per cent will be due to the adjustment applied in the previous iteration. As a result, any weight adjustment will have a continuing effect, following an exponential decay.

The 'smoothing out' effect of this process is the main benefit of the momentum parameter, since it prevents outliers from forcing learning in an undesirable direction. By using a momentum, weight changes in the training of the NN are channelled in the same direction of the preceding iteration. This is particularly true when higher momentum values are used. In such a case, high momentum tends to accelerate convergence, giving it, as in the word, 'momentum' (Hagan et al. 1996). Alternatively, lower momentum values may be suitable for data which are more regular or smoother, or when the functional relationships to be learned are relatively simple. Generally, experimenting with different values of momentum might be necessary, as in the case of LR, in order

[5] The momentum parameter cannot exactly assume the value 1. The reason for this *caveat* is easily shown by an example. If the momentum was set at 1, 100 per cent of the previous error adjustment would be used at each stage of the training. Because no previous adjustments are present at the very first training iteration, the first weight adjustment would be 0. But the same adjustment (0) would be repeated at each iteration, since the current error is not considered, resulting in no training whatsoever.

to find the appropriate value for the problem at hand, unless more sophisticated methods are employed in order to determine the right momentum value (see, for example, Yu et al. 1995). These methods can also be linked to the use of adaptive LRs.

4.2.2 Sensitivity Analysis

When testing for values of LR and momentum, an exhaustive search of the (0, 1) interval for both parameters, including all their possible combinations, would be rather time-consuming. Sharda and Patil (1992) suggest a simpler strategy, based on the use of three values (0.1, 0.5, 0.9) for each parameter. The resulting nine combinations can be tested separately, with no excessive computation efforts, while covering most of the spectrum of possible values.

The same approach is followed in our experiments. The sensitivity of the NN models to different LR and momentum values is tested for the above nine combinations of values, using always a sigmoid (logistic) activation function. We choose for testing the baseline model presented in Section 3 (Model B), because of its simple application and stable performance seen in previous experiments. For all combinations of LR and momentum, and for West and East models, the ideal training time is identified by means of early stopping (see Section 2). Table 1 shows the pooled MSE and MAPE obtained for the years 2001, 2002, 2003, and 2004. The computation of the pooled error increases the reliability of our statistical findings, by averaging out the stochastic variability of the models' single-period application.

Table 1 – Sensitivity analysis for learning rate and momentum: Model B, West and East Germany, years 2001–04

West Germany							
MSE (/1000)				MAPE			
Learning rate / Momentum	0.1	0.5	0.9	Learning rate / Momentum	0.1	0.5	0.9
0.1	10242.45 (6)	9481.17 (3)	10072.73 (5)	0.1	3.72 (4)	3.65 (2)	3.79 (6)
0.5	*9226.85 (1)*	9575.08 (4)	9478.07 (2)	0.5	*3.59 (1)*	3.73 (5)	3.70 (3)
0.9	12161.96 (9)	10962.08 (7)	11839.88 (8)	0.9	4.04 (8)	3.83 (7)	4.11 (9)
East Germany							
MSE (/1000)				MAPE			
Learning rate / Momentum	0.1	0.5	0.9	Learning rate / Momentum	0.1	0.5	0.9
0.1	*2391.72 (1)*	3609.21 (8)	3786.61 (9)	0.1	3.46 (4)	3.44 (3)	3.46 (5)
0.5	3248.86 (6)	3026.33 (4)	2891.25 (2)	0.5	3.43 (2)	3.72 (7)	3.73 (8)
0.9	2938.76 (3)	3206.21 (5)	3305.50 (7)	0.9	*3.42 (1)*	3.86 (9)	3.70 (6)

Note: The ranking of the NN models is shown in brackets.

In Table 1, the stochastic variability that is inherent to NNs generates different degrees of statistical performance for the West and East German NN models, and for the two error indicators used. However, combinations of low LR and medium momentum (0.1, 0.5) seem to provide lower statistical error.

Our finding is that a low LR, matched with a medium-range momentum, leads to better performance for the case of regional employment forecasts. A NN employing such parameters is expected to show a potentially slower convergence (indeed, a high number of iterations is needed for NN training), but at the same time to experience more stable learning behaviour between iterations. The medium value for the momentum parameter (0.5) allows for a lasting effect of the learning obtained at each step.

Our results can be compared with the ones by Tang et al. (1991), who found that low LR (and higher momentum) values are adequate for use with complex data (while higher LRs are appropriate for simpler data). Whether or not our findings match these considerations relies on whether our data should be considered 'complex'. Generally, Tang and Fishwick (1993) state that,

for each series of data, a set of NN parameters can be found which performs significantly better than the rest. This consideration stresses once again the crucial role played by the learning parameters in the performance of NNs. The inconsistent results in the literature regarding the search of ideal values of the learning parameters (see, for example, Chakraborty et al. 1992; Sharda and Patil 1992) are blamed by Zhang et al. (1998) to the minimum search inefficiencies of the BPA.

4.2.3 Adaptive learning rate

We pointed out in Section 3 that the BPA has flaws, that is, it can have slow convergence (if any) (Kuan and Hornik 1991) and, most importantly, can get trapped in local minima. Several techniques have been developed in order to solve the problem of slow convergence of the BPA. The BPA is also sensitive to the initial conditions chosen, and can show oscillations in the computation units' output (Sarkar 1995). While the momentum parameter can be seen as – and mostly is – a regulator of the oscillation and local minima problems in the BPA (and involving the LR parameter), its value is chosen *a priori*, and is therefore not tied to the actual progress of the NN iterations.

In order to overcome these limitations, the use of adaptive learning rate (ALR) has been proposed. In the *bold driver* method (Vogl et al. 1988), the LR – as defined in Section 4.2.1 – is augmented by a factor ρ when the error computed at iteration *n* is greater than the one previously found at iteration *n* – 1. Otherwise, the LR is diminished by a factor σ when the error decreases.[6] A further step in the application of ALR techniques is the implementation of NNs that have *multiple* ALRs. In the *self-adaptive backpropagation (SAB)* method, each weight can have its own LR, which is computed as the partial derivative of the learning error estimator. The method is based on the idea that the same LR may not be appropriate for all the weights of the NN. Moreover, in the *SuperSAB* method, it is suggested that the ρ and σ factors that modify the multiple LRs should be also different in value, and that the decrease in the LR caused by the σ factor should be greater (see Jacobs 1988; Tollenaere 1990). Tollenaere suggests that the SuperSAB algorithm considerably speeds up learning.

The ALR approaches listed above provide a somehow faster learning for NNs. On the other hand, Park et al. (2000) advise that these methods can not completely avoid the algorithm from stalling in slow convergence plateaus. This is because this class of methods uses the same search direction that is used in the conventional BPA. Consequently, we want to test if an ALR approach can provide improved statistical performance in comparison with the one of the NNs with fixed LR. We consider two NN models, based on Model B, for the years 2001–04: the first model employs a LR of 0.1, while the second model uses an ALR. Both models have a momentum of 0.5, as found in Section 4.2.2). Again, a sigmoid activation function is used in both models.

With regard to the implementation of the ALR used, this is implemented as follows:

- The LR is modified at this training iteration. The extent of its recalculation is based on the error computed at the previous iteration;
- If the error decreases as a result of the last iteration, the LR drops proportionally to the error decrease. If the error increases, the LR also increases proportionally.
- The training of the NN models ends once the stopping condition is satisfied.

Our first question is if the ALR algorithm provides, in our case, a faster convergence, which requires us to observe the evolution of the training error. When plotting the error against the number

[6] The momentum parameter can also be modified: that is, forced to 0 when the error increases, and brought back to its value in the opposite case (Hagan et al. 1996). In addition, Yu et al. (1995) propose a dynamically adaptive method for the optimization of the LR, which employs derivative information. Alternatively, Plagianakos (1999) suggests an acceptability criterion for the modification of the LR, based on the previous *M* computed errors. This approach appears to speed up convergence of the NNs and to make them more robust against oscillations.

of training epochs, the NNs with an ALR seem to reach a stable training error (converge) faster than the ones with fixed LR (Figure 1). This 'informal' result is consistent with the literature.

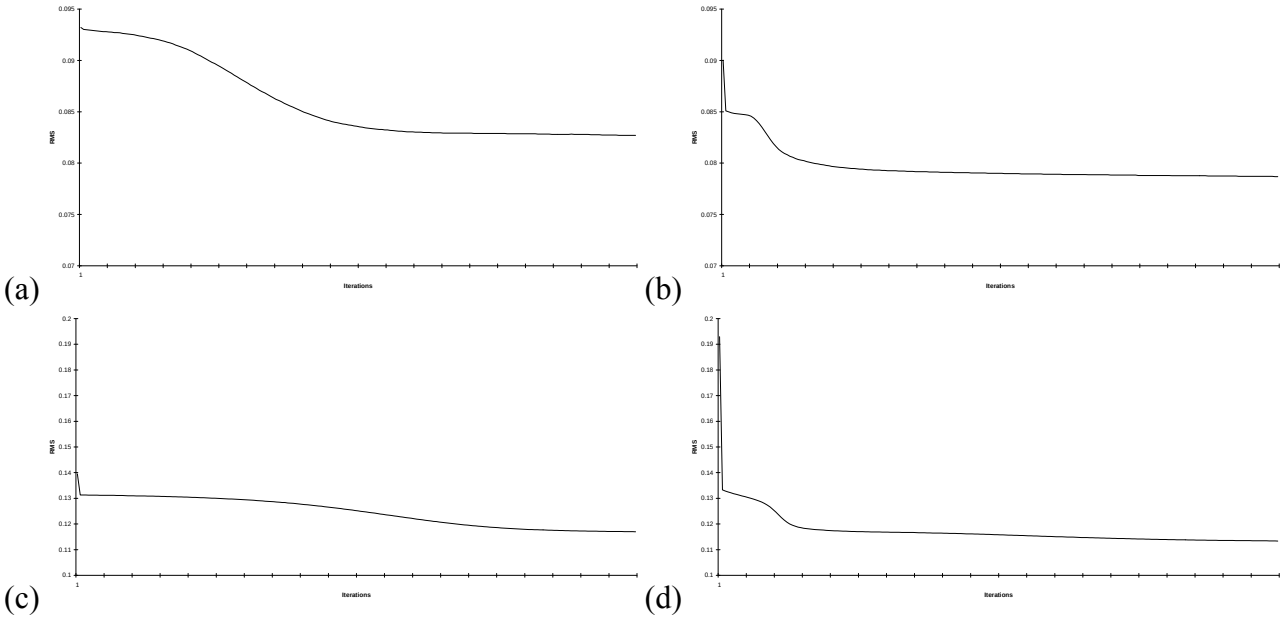

Figure 1 – Training error (RMS) evolution over 400 iterations, for West and East German NN models: West Germany, fixed LR (a), ALR (b); East Germany, fixed LR (c), ALR (d)

The subsequent question is, therefore, if the algorithm can improve the statistical performance of the models. Table 2 reports the error obtained in the simulated out-of-sample forecasts for the conventional fixed LR models, as well as of the ALR models, and shows a similar statistical performance for the fixed and adaptive LR models compared. This result is found for both data sets, in particular for East Germany; the differences in the statistical error can be considered of limited relevance, when compared with the variability shown in the LR/momentum and activation function analyses.

Table 2 – Sensitivity analysis for adaptive learning rate: Model B, West and East Germany, years 2001–04

	West Germany		*East Germany*	
	MSE (/1000)	MAPE	MSE (/1000)	MAPE
Fixed LR (0.1)	*9226.85 (1)*	*3.59 (1)*	3248.86 (2)	*3.43 (1)*
Adaptive LR	9670.04 (2)	3.75 (2)	*3229.53 (1)*	3.45 (2)

Note: The ranking of the NN models is shown between brackets.

However, to gain inferential evidence regarding the performance of our competing specifications, the models can be further compared by using a forecast equality non-parametric test, the sign test (ST) (Lehmann 1998). The ST is based on the following idea: if two models, Model 1 and Model 2, are equally accurate, the number of forecasts of Model 2 which have a bigger error than Model 1 are expected to be 50 per cent of the total number of forecasts obtained. Consequently, Model 1 will be considered superior to Model 2 if Model 2 has higher forecasting errors in more than 50 per cent of the cases. The ST statistic is computed as:

$$\mathrm{ST} = \frac{C - \frac{N}{2}}{\frac{1}{2}\sqrt{N}}, \tag{2}$$

where C is the number of times Model 2 shows higher errors than Model 1, and N is the number of forecasts carried out (in our case, the number of districts concerned times the number of years forecasted). The ST statistic follows a normal distribution $N(0, 1)$, while the null hypothesis H_0 is of equality of the forecasting models.

We combine the error obtained for the four years of simulated out-of-sample forecasts, 2001, 2002, 2003, and 2004, obtaining 1304 forecasting errors for West Germany, and 452 for East Germany. Comparing the ALR models (Model 1) to the fixed-LR models (Model 2), we obtain ST statistic values of –7.26 and –3.48 for the West and the East, respectively, suggesting that the fixed-LR NN models should be preferred to the ALR NN models.

On the basis of the analyses carried out in this section, we can conclude that, in our forecasting experiments, ALR does not provide relevant approximation advantages, if only faster convergence of the algorithm, consistently with the literature. However, it should be pointed out that such a result may be greatly relevant when computational issues arise.

4.3 Activation Function

4.3.1 Description

The greater benefit of using NNs is their nonlinear behaviour, which allows them to approximate nearly every type of function. Nonlinearities are introduced in NNs by means of the activation function. Ideally, any differentiable function can be used as an activation function. Practically, only a few nonlinear functions are usually considered for NNs, that is:

- sigmoid (logistic) functions;
- augmented ratio functions;
- Gaussian functions; and
- hyperbolic (tangent) functions.

As a special case, we also consider:

- linear functions,

the use of which is sometimes suggested in NNs (see below). The sigmoid function is, anyway, the most widely used activation function. It is a smooth function, which returns nearly proportional outputs for intermediate values, while smoothing out values at the extremes of the spectrum. The augmented ratio function and the hyperbolic function are mostly similar to the sigmoid function, but, in the augmented ratio function, small values are rounded to 0, while the hyperbolic function is negatively oriented, tending to force extreme values of the distribution to either 1 or –1. The Gaussian function forces small values to 1, and extreme values to 0. The augmented ratio function looks like an inverted Gaussian function. Differently from the functions described above, a linear function proportionally rescales the values within the (0, 1) interval.

While any of the described functions can be implemented in NNs, there is no clear rule on how to select the most appropriate activation function. Some heuristic rules have been proposed in the literature in order to select a suitable function, such as in Klimasauskas (1991). The author suggests the use of sigmoid functions for classification problems (for example, with binary outputs), and of hyperbolic functions for forecasting problems, that is, when learning about deviations from the average is involved. A different function can ideally be used for each computational unit in the NN (for example, both linear and sigmoid functions, as in Wong 1991). While the usual NN models found in the literature employ the same activation function for all units, examples can also be found of NNs in which a different function is selected for the output units. Sigmoid functions are mostly used in the input and hidden layers, while there is no agreement on what activation function should

be employed for the output units. In this latter regard, Zhang et al. (1998) and Rumelhart et al. (1995) suggest the use of linear functions. Zhang et al. cite a set of studies following the same procedure (see, for example, Srinivasan et al. 1994; Kuan and Liu 1995), which, according to the authors, provide no clear results on whether linear or nonlinear activation functions should be preferred for the implementation in the output units. As an additional *caveat*, it is outlined that NNs with linear output units are not able to approximate data with trends (Cottrell et al. 1995). This last aspect is less relevant in our case, as the NN models developed here employ growth rates.

4.3.2 Sensitivity Analysis

A sensitivity analysis of the performance of NNs with different activation functions would ideally require a full exploration of the possibilities available, and also of the mixed approaches discussed above. In this paper, we are limited to testing NNs employing the same activation function for all layers of units.[7]

The activation functions that are tested here are: (1) sigmoid; (2) augmented ratio; (3) Gaussian; (4) hyperbolic; and (5) linear, as outlined in the previous section. While the linear function is normally used in the output layer only (see preceding section), our experiments intend to test its implementation in a whole NN. All models are based on the baseline model discussed above (Model B, see Section 3), employ the choice of learning parameters (a LR of 0.1 and a momentum of 0.5) seen above in Section 4.1.2, and are carried out for the years 2001, 2002, 2003, and 2004. Table 3 presents the results obtained for both West and East German models. Statistical error is computed as MSE and MAPE.

Table 3 – Sensitivity analysis for activation functions: Model B, West and East Germany, years 2001–04

West Germany	*Sigmoid*	*Aug. Ratio*	*Gaussian*	*Hyperbolic*	*Linear*
MSE (/1000)	*9226.85 (1)*	9297.49 (2)	10131.27 (4)	9945.25 (3)	12307.48 (5)
MAPE	*3.59 (1)*	3.68 (3)	3.71 (4)	3.66 (2)	4.07 (5)
East Germany	Sigmoid	Aug. Ratio	Gaussian	Hyperbolic	Linear
MSE (/1000)	3248.86 (3)	3678.93 (5)	2653.34 (2)	3315.57 (4)	*2505.84 (1)*
MAPE	3.43 (3)	3.44 (4)	*3.41 (1)*	3.42 (2)	3.73 (5)

Note: The ranking of the NN models is shown between brackets.

The statistical results shown in Table 3 generally confirm, in particular for the West German NN models, the results found in the literature: the NNs models employing a sigmoid activation function show stable and good statistical performance. This finding follows in the line of the general consensus on the use of the sigmoid function, and confirms our initial choice of activation function (see Section 3). More generally, the performance of all the nonlinear functions – for the West and the East – appears to be rather homogeneous in terms of MAPE. With regard to the NN models for East Germany, we note that the linear activation function appears to provide the best statistical result when the MSE is considered (while its results for West Germany are not satisfactory). This finding suggests a possible tendency towards linearity of the East German data.

While the full reasons leading to the differences in the performance of the linear function should be further investigated, in order to better grasp the relationship between data complexity and the ideal (linear or nonlinear) approximation function to use, we again use the sign test (ST) in order to find a winning model with regard to East Germany. We test the equality between the NN model employing a Gaussian activation function and the baseline sigmoid NN model. The ST statistic of –3.76 suggests that the baseline model, based on a Sigmoid function, is preferable.

In summary, on the basis of our results, we might suggest that the sigmoid activation function should be used. However, more in-depth explorations should be carried out in the light of the mixed results of the linear activation function, and in the framework of alternative multi-function NN

[7] The software used for our experiments does not allow to select multiple simultaneous functions.

specifications. Finally, the statistical results of the sensitivity analysis carried out above call for further testing, in particular to verify how different model specifications (in terms of input variables) may lead to varying performance once the NN settings selected in this section are in place. This analysis is provided in the next section.

5 Evaluation of Different Neural Network Model Specifications

In the light of the findings of the sensitivity analysis carried out above, which allow us to select a set of learning parameters and an activation function, we want to evaluate the statistical performance of different NN model specifications exploiting the findings of Section 4. Table 4 presents the statistical results computed, for the six NN models presented earlier on in Section 3, for the usual four forecasting periods: 2001, 2002, 2003, and 2004. The LR and momentum parameter values used are 0.1 and 0.5, respectively, while a sigmoid activation function is employed.

Table 4 – Pooled statistical error of the NN models; West and East Germany, years 2001–04

West	*MSE (/1000)*	*MAPE*	*East*	*MSE (/1000)*	*MAPE*
Model B	27474.58 (3)	5.67 (3)	Model B	3248.86 (2)	3.43 (5)
Model BD	25983.19 (2)	5.10 (2)	Model BD	*2543.62 (1)*	3.01 (2)
Model BSS	29384.08 (4)	5.85 (4)	Model BSS	13633.35 (6)	*2.86 (1)*
Model BSSN	41228.08 (5)	7.18 (5)	Model BSSN	8080.81 (3)	3.63 (6)
Model BSSR	55694.54 (6)	7.78 (6)	Model BSSR	8676.52 (5)	3.31 (4)
Model BW	*12749.12 (1)*	*4.29 (1)*	Model BW	8659.66 (4)	3.19 (3)

Note: The ranking of the NN models is shown in brackets.

The statistical results shown in Table 4 can be read as follows: for West Germany, (a) the inclusion of information on the district classification (Model BD) and wages (Model BW) appears to improve the forecasting potential of the NN models, as the baseline Model B follows closely, while (b) the shift-share-enhanced models (SS-models) do not lead to better statistical performance; for East Germany, our results seem more unclear: (c) Model BD minimizes the MSE indicator, while Model BSS does the same for MAPE (but has rather high MSE!). Consequently, Model BD appears to minimize the effect of squared large forecasting errors in the MSE formula. On the other hand, Model BSS minimizes the average percentage error.

In order to cope with the contrasting statistical evidence of Table 4, we again resort to the use of forecast equality tests, viz. the sign test (ST). With regard to the NN models developed for West Germany, we test whether Model BW (employing as an additional input the variation of average daily wages) outperforms our baseline model (Model B). The normally-distributed test statistic obtained for is –26.42, showing that the Model BW, while minimizing the average error (both squared and percentage), is outperformed by the baseline model for most forecasts. With regard to the NN models of East Germany, we test whether Model BD, which has both low MSE and MAPE, outperforms the baseline model. The test statistic result (2.26) suggests, with a 95 per cent confidence level, that indeed Model BD is preferable to the baseline (outperforming the baseline in 250 of 452 total cases).

Overall, our results suggest that the baseline model (Model B) and the district-type model (Model BD) are most suitable for our research problem, given the learning parameters and activation function chosen during the sensitivity analysis. With regard to the interpretation of these findings, attention should be focused – with the due caution – on the use of the socio-economic covariates. It is interesting to note that the use of variables such as wages and urbanization level does not unequivocally improve the results, suggesting a overall – but logical – predominance of the autoregressive effects in the determination of employment growth rates. Similarly, the inclusion of shift-share components (conventional, spatial, and regression shift-share) appears to increase the

computational complexity of the models (nine new variables are included, as many as the sectors considered), without increasing the forecasting reliability of the NN models. This result confirms the problem of finding out which region-specific information is relevant for a specific case.

Finally, we should point out that the parameters chosen for our NN models might not fit all model specifications, since they were tested specifically on Model B only. In other words, more investigation is needed with regard to the influence of the new parameters on NN models employing more or less rich data. It could be argued that, in order to improve the performance of alternative models, such as Model BSS (which is richer in information), specific sensitivity analyses should be carried out, which might lead to different conclusions. Nonetheless, our analysis offers an overview of the possible steps to be taken in this direction.

6 Conclusions

In this paper we presented a sensitivity analysis of the performance of NN models developed for regional employment forecasting. Because of the regional focus, the NN models developed here employ panel data, rather than time series. Our experiments focused on two aspects of NNs. In the first phase of the experiments, we carried out a sensitivity analysis, starting from a baseline model, investigating the effect of varying learning parameters and functional forms on the NN models' forecasting performance. In doing so, we tested all different NN settings by simulating out-of-sample forecasts for four time periods. In the second phase, we implemented five additional models utilizing different inputs, and we evaluated their statistical performance in the light of the sensitivity analysis findings, and for the same time periods.

The sensitivity analysis carried out concerned the investigation of different NN parameters, and their influence on the forecasting performance of our NN models. Our analyses show that, for the basic case of Model B, low learning rate (LR) values and medium momentum values tend to improve the forecasts of our models. On the other hand, the tests carried out on adaptive learning rate (ALR) suggest that no forecasting benefit would be gained from the use of adaptive parameters, except for a quicker training of the NNs. Finally, our analysis found that the sigmoid (logistic) function conventionally used in NN models is appropriate for the forecasting problem concerned, although the results obtained for the linear activation function suggest that it might be suitable for the case of East Germany (where the employment trends appear to be less complex). This result calls for much needed testing on the linearity of the employment data, in particular for East Germany.

As a last undertaking, we evaluated, on the basis of the findings of the sensitivity analysis, a set of five additional NN models, which introduced further input variables, ranging from an indicator of urbanization/agglomeration to shift-share analysis (SSA) components of various derivation, to regional wages variation. We observed varying levels of statistical error, for both the West and East Germany models. We finally identified two preferred model specifications, viz. Model B for West Germany, and Model BD for East Germany. While the baseline model (on which the sensitivity analysis was carried out) is most suitable for the West, the introduction of an indicator variable identifying varying levels of urbanization and agglomeration appears to significantly improve the forecasting power of the NN models for the East. With regard to the NN models employing SSA-inspired variables, no significant gain was obtained, likely because of different levels of computational complexity and richness of information, which might require new targeted sensitivity analyses in order for them to be fully exploited.

The analyses illustrated in the present paper can be expanded by carrying out further research in several directions. From an empirical viewpoint, a longer data span (for example, by obtaining newer data) would allow us to increase the number of testing years and, consequently, the reliability of the average (pooled) statistical results. The development of further NN models, utilizing new variables (such as unemployment or migration) is also desirable. Additionally, the sensitivity

analysis carried out in Section 4 would benefit from being extended to more model specifications. Finally, the contrasting evidence of the statistical indicators used in the evaluation of the forecasting performance, and the results of the sign tests (ST) should be investigated further, in particular in relation to the analyst's specific objectives, whether these are minimizing the number of outliers in the models' errors, maximizing the number of forecasts showing the right sign, or other particular targets.

From a methodological point of view, it might be desirable to test out more elaborate NN models, such as time-delay NNs (Waibel et al. 1989), or multi-function NNs. In particular, the testing of linear functions integrated within NNs, as discussed in Section 4.3, should be a main objective. Also, a more in-depth analysis of the spatial interactions among districts might help understanding better the regional phenomena. The incorporation, in Model BSSN, of information on the (employment) performance of the 'neighbours' was a first step in this direction. In this framework, the potential of methods such as spatial filtering (Griffith 2003) for developing spatial/regional variables should also be considered.

References

Baltagi B.H. (2001) *Econometric Analysis of Panel Data* (2nd ed.). Wiley: Chichester New York

Böltgen F. and E. Irmen (1997) Neue Siedlungsstrukturelle Regions- und Kreistypen. *Mitteilungen und Informationen der BfLR* H. 1, S. 4-5, (in German)

Chakraborty K., K. Mehrotra, C.K. Mohan and S. Ranka (1992) Forecasting the Behavior of Multivariate Time Series Using Neural Networks. *Neural Networks* 5 (6), 961-70

Cooper J.C.B. (1999) Artificial Neural Networks versus Multivariate Statistics: An Application from Economics. *Journal of Applied Statistics* 26, 909-21

Cottrell M., B. Girard, Y. Girard, M. Mangeas and C. Muller (1995) Neural Modelling for Times Series: A Statistical Stepwise Method for Weight Elimination. *IEEE Transactions on Neural Networks* 51 (2), 240-54

Dunn E.S. (1960) A Statistical and Analytical Technique for Regional Analysis. *Papers and Proceedings of the Regional Science Association* 6, 97-112

Gopal S. and M.M. Fischer (1996) Learning in Single Hidden Layer Feedforward Neural Network Models: Backpropagation in a Spatial Interaction Modeling Context. *Geographical Analysis* 28 (1), 38-55

Gorr W.L., D. Nagin and J. Szczypula (1994) Comparative Study of Artificial Neural Network and Statistical Models for Predicting Student Grade Point Averages. *International Journal of Forecasting* 10 (1), 17-34

Griffith D.A. (2003) *Spatial Autocorrelation and Spatial Filtering: Gaining Understanding through Theory and Scientific Visualization*. Springer: Berlin New York

Hagan M.T., H.B. Demuth and M.H. Beale (1996) *Neural Network Design*. PWS Pub.: Boston

Jacobs R.A. (1988) Increased Rates of Convergence Through Learning Rate Adaptation. *Neural Networks* 1 (4), 295-308

Klimasauskas C.C. (1991) Applying Neural Networks. Part 3: Training a Neural Network. *PC/AI Magazine* 5, 20-4

Kuan C.-M. and T. Liu (1995) Forecasting Exchange Rates Using Feedforward and Recurrent Neural Networks. *Journal of Applied Econometrics* 10 (4), 347-64

Kuan C.M. and K. Hornik (1991) Convergence of Learning Algorithms with Constant Learning Rates. *IEEE Transactions on Neural Networks* 2, 484-8

Lehmann E.L. (1998) *Nonparametrics: Statistical Methods Based on Ranks* (rev. ed.). Prentice Hall: Upper Saddle River

Nazara S. and G.J.D. Hewings (2004) Spatial Structure and Taxonomy of Decomposition in Shift-Share Analysis. *Growth and Change* 35 (4), 476-90

Park H., S.-I. Amari and K. Fukumizu (2000) Adaptive Natural Gradient Learning Algorithms for Various Stochastic Models. *Neural Networks* 13, 755-64

Patuelli R., A. Reggiani, P. Nijkamp and U. Blien (2006) New Neural Network Methods for Forecasting Regional Employment: An Analysis of German Labour Markets. *Spatial Economic Analysis* 1 (1), 7-30

Patuelli R., S. Longhi, A. Reggiani and P. Nijkamp (2007) Forecasting Regional Employment in Germany by means of Neural Networks and Genetic Algorithms. *Environment & Planning B* 35 (4), 701-22

Plagianakos V.P., M.N. Vrahatis and G.D. Magoulas (1999) Nonmonotone Methods for Backpropagation Training with Adaptive Learning Rate. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Washington (Vol. 3), pp. 1762-7

Rosenblatt F. (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review* 65, 386-408

Rumelhart D.E. and J.L. McClelland (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press: Cambridge

Rumelhart D.E., R. Durbin, R. Golden and Y. Chauvin (1995) Backpropagation: The Basic Theory. In Y. Chauvin and D.E. Rumelhart (eds), *Backpropagation: Theory, Architectures, and Applications*. Lawrence Erlbaum Associates: Hillsdale, pp. 1-34

Sarkar D. (1995) Methods to Speed Up Error Back-Propagation Learning Algorithm. *ACM Computing Surveys* 27 (4), 519-42

Schintler L.A. and O. Olurotimi (1998) Neural Networks as Adaptive Logit Models. In V. Himanen, P. Nijkamp and A. Reggiani (eds), *Neural Networks in Transport Applications*. Ashgate: Aldershot Brookfield, pp. 131-60

Sharda R. and R.B. Patil (1992) Connectionist Approach to Time Series Prediction: An Empirical Test. *Journal of Intelligent Manufacturing* 3 (5), 317-23

Srinivasan D., A.C. Liew and C.S. Chang (1994) A Neural Network Short-term Load Forecaster. *Electric Power Systems Research* 28, 227-34

Tang Z. and P.A. Fishwick (1993) Feedforward Neural Nets as Models for Time Series Forecasting. *INFORMS Journal on Computing* 5 (4), 374-85

Tang Z., C. Almeida and P.A. Fishwick (1991) Time Series Forecasting Using Neural Networks vs Box-Jenkins Methodology. *Simulation* 57 (5), 303-10

Tollenaere T. (1990) SuperSAB: Fast Adaptive Back Propagation with Good Scaling Properties. *Neural Networks* 3 (5), 561-73

Vogl T.P., J.W. Mangis, A.K. Rigler, W.T. Zink and D.L. Alkon (1988) Accelerating the Convergence of the Back-Propagation Method. *Biological Cybernetics* 59, 257-63

Waibel A.H., T. Hanazawa, G.E. Hinton, K. Shikano and K.J. Lang (1989) Phoneme Recognition Using Time-delay Neural Networks. *IEEE Transactions on Acoustics, Speech and Signal Processing* 37 (3), 328-39

Weisstein E.W. (2006) *Method of Steepest Descent, from MathWorld*, from http://mathworld.wolfram.com/MethodofSteepestDescent.html

Werbos P. (1974) *Beyond Regression: New Tools for Predicting and Analysis in the Behavioral Sciences.* Unpublished PhD thesis, reprinted by Wiley & Sons, 1995, Harvard University

Wong F.S. (1991) Time Series Forecasting Using Backpropagation Neural Networks. *Neurocomputing* 2 (4), 147-59

Yu X.H., G.A. Chen and S.X. Cheng (1995) Dynamic Learning Rate Optimization of the Backpropagation Algorithm. *IEEE Transactions on Neural Networks* 6 (3), 669-77

Zhang G., B.E. Patuwo and M.Y. Hu (1998) Forecasting with Artificial Neural Networks: The State of the Art. *International Journal of Forecasting* 14 (1), 35-62