J Geogr Syst (2011) 13:67–85
DOI 10.1007/s10109-010-0133-5

ORIGINAL ARTICLE

Neural networks for regional employment forecasts: are the parameters relevant?

Roberto Patuelli · Aura Reggiani · Peter Nijkamp · Norbert Schanne

Received: 26 February 2009 / Accepted: 18 August 2010 / Published online: 14 September 2010

Abstract In this paper, we present a review of various computational experiments concerning neural network (NN) models developed for regional employment forecasting. NNs are nowadays widely used in several fields because of their flexible specification structure. A series of NN experiments is presented in the paper, using two data sets on German NUTS-3 districts. Individual forecasts are computed by our models for each district in order to answer the following question: How relevant are NN parameters in comparison to NN structure? Comprehensive testing of these parameters is limited in the literature. Building on different specifications of NN models—in terms of explanatory variables and NN structures—we propose a systematic choice of NN learning parameters and internal functions by means of a sensitivity analysis. Our results show that different combinations of NN parameters provide significantly varying statistical performance and forecasting power. Finally, we note that the sets of parameters chosen for a given model specification cannot be light-heartedly applied to different or more complex models.

R. Patuelli (✉)
Institute for Economic Research (IRE), University of Lugano, Lugano, Switzerland
e-mail: roberto.patuelli@usi.ch

R. Patuelli
The Rimini Centre for Economic Analysis, Rimini, Italy

A. Reggiani
Department of Economics, University of Bologna, Bologna, Italy
e-mail: aura.reggiani@unibo.it

P. Nijkamp
Department of Spatial Economics, VU University Amsterdam, Amsterdam, The Netherlands
e-mail: pnijkamp@feweb.vu.nl

N. Schanne
Institute for Employment Research (IAB), Nuremberg, Germany
e-mail: Norbert.Schanne@iab.de

Keywords Neural networks · Sensitivity analysis · Employment forecasts · Local labour markets

JEL Classification C45 · E27 · R23

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: (1) the imbalance between the increased number of regions to forecast for and the time span of the observations available; and (2) 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 that 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 socioeconomic 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 specification 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. 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—and sometimes complementary—model specifications in an NN context so as to test the robustness of forecasting results. Another caveat regarding 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).

In this context, the objective of the present paper is to investigate the role of parameters in NN models—developed for regional employment forecasts—by means of a sensitivity analysis. Studies in this respect are in fact rare, and mostly related to different topics than regional forecasting (an exception being Gopal and Fischer 1996). More specifically, we aim to answer the following question: How relevant are NN parameters in comparison to NN structure? We use German

regional employment variables as a case study, and develop and estimate a set of NN models.

The paper is structured as follows: Sect. 2 provides a brief pedagogical description of the working of NNs. Section 3 illustrates a set of NNs models developed for regional employment forecasting and the results obtained for different NN structures tested. Next, Sect. 4 presents a sensitivity analysis carried out to test different combinations of learning parameters and internal functional forms, while Sect. 5 reviews the NN structure and parameters findings obtained and offers an evaluation and comparative discussion of the NN models' statistical performance. Finally, Sect. 6 draws methodological and empirical conclusions, as well as suggestions for future research.

2 Neural networks

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. Though they are often referred to, in particular in social sciences, as a 'black box' approach 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).

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).[1] In feedforward NNs, every unit is connected to all units in the successive layer, and connections only go 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 (Fischer 2001, p. 23):

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

where $\mathbf{u}_{n-1} = \{u_{1,n-1}, \ldots, u_{k,n-1}\}$ 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 Sect. 4.3), while the latter aggregates the information processed by the units of the preceding layer (in Eq. 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 $w_{ij,n-1}$ are recursively computed during the

[1] A NN with no hidden layers is called a one-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 two-layer structure, and so on.

'training' of the NN, and they represent the 'knowledge' generated by the NN. The backpropagation algorithm (BPA, Rumelhart and McClelland 1986) is the algorithm most commonly used for the computation of the above weights. The learning process of the NN is given by the comparison between the output generated from Eq. 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 readjustments of the weights,[2] until a stopping condition is satisfied.

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 occur when excessive iterations are carried out, a situation that may be detected by observing deterioration in the statistical error of the NN. A number of techniques can 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 reaches a slow convergence or increases. NNs were also shown to be sensitive to changes in their structure, 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 the following sections (Sect. 3 with regard to the choice of an NN structure and Sect. 4 with regard to the optimal NN parameters and functional specifications).

3 Neural networks for forecasting regional employment: a review of specifications and internal structures

The variable we aim 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, forecasting 2 years ahead $(t, t + 2)$, and use panel data for the periods 1987–2004 and 1993–2004, for West and East Germany, respectively.[3] The panel nature of the data 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?

In addition to the inclusion of time-autoregressive effects in our NN models, obtained by employing as input variables the 2-year lagged sectoral employment variations, we aim to capture year-specific shocks, so to purge inference from anomalous employment variations due to exogenous events at the aggregate level

[2] 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] 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). The employment data refer, for each year, to the second quarter.

(such as a recession period). We may capture such effects in our models following two alternative approaches. The first approach (henceforth, A-type models) consists of the use of yearly dummy variables. Each of the dummy variables enters the NN model separately and consequently influences only NN training for the corresponding year (having value zero for each other year). This approach may be compared to what in panel econometrics is referred to as 'time fixed effects' (or more generally as a 'factor' in statistics). The time dummies, once entered as inputs in the NN, have a nonlinear effect, in the same fashion as all other covariates. The second feasible approach (henceforth, B-type models) is to employ a variable that identifies—by means of a text (string) variable—the years concerned. This approach is made possible by internally rescaling the text variable, that is, each year is associated with a numerical value within the (0,1) interval, therefore identifying year-specific intercepts. Similarly to the first approach, we generically compare this solution to so-called 'time random effects', although normality of the rescaled values is not guaranteed in this case.

The suitability and statistical performance of the two proposed solutions to the incorporation of time-specific effects have been recently tested in Patuelli et al. (2008). They test both approaches on a similar dataset of German employment, while controlling in parallel for the inclusion of alternative sets of covariates and for the subjective or genetic-algorithm-based determination of NN structures and parameters. Patuelli and coauthors find that the two approaches tend to minimize different statistical error indicators (MSE and MAPE, respectively) and that forecast equivalence tests suggest that A-type models—based on time dummies—should be preferred. However, this result appears to be quite sensitive to the forecasting year chosen. When forecasts obtained over a higher number of forecasting years are pooled together (Patuelli et al. 2006a), the B-type models—based on the rescaled time variable—are preferred. In addition, the B-type models may be more convenient for future temporal expansions of the models: in fact, A-type models require the inclusion of additional dummy variables when new training years are added, which altogether changes the structure of the NN (for example, a 10-1-1 NN model would acquire a 12-1-1 structure if 2 years of data were added). Such changes would modify and eventually require a new search for the ideal NN structure and parameters. On the basis of the results in the literature, we choose to retain only models of type B in our analysis.

In addition to the time approaches discussed above, we employ, as the main covariates in all models, the growth rates observed in full-time employment, for the period $(t-2, t)$, in order to include autoregressive effects. We subdivide the employees in nine sectors, ranging from primary goods to services.

We extend this baseline model (hereforth, Model B) by means of additional covariates, therefore defining five more models. Two models employ additional basic information about district characteristics and average daily wages:

- Model BD uses a nine-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 uses information on average regional daily wages of full-time workers. The wage variable aims to capture the well-known relationship between labour supply/demand and wages.[4]

Three additional models (NN-SS models) represent a further advancement in forecasting employment by means of NNs. We augment Model B with components derived from several shift-share analysis (SSA) approaches proposed in the literature. The related models are illustrated in Patuelli et al. (2006b) as follows:

- Model BSS uses the competitive effect components computed by means of 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, 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. (2006b). The new effects ought to be a fine-tuning of the ones used in Model BSS.

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).[5] Patuelli et al. (2006a, b) assess the statistical performance of each model described above. In particular, in order to find the most suitable NN structure, they test in each case: (1) a one-layer structure; (2) two-layer structures with 5, 10, or 15 hidden units; and (3) a three-layer structure with five hidden units in both hidden layers.[6] During this phrase, 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

[4] The level of geographical aggregation chosen (NUTS-3) leads us to examine areas smaller than functional areas. Consequently, models including wages as input data might be more properly re-estimated at a larger geographical scale, at which public subsidies are evaluated. Nevertheless, local policy makers (operating at the district level) may want to obtain forecasts at their level of jurisdiction. Additional explanatory variables, such as agglomeration/urbanization, and shift-share components may help to partially account for this issue, which calls for further investigation in future research.

[5] Attempts at estimating unique NN models for the entire set of German districts proved unsuccessful, suggesting different autoregressive effects for the West and East German districts, which can be due, for example, to the widely different economic structures of the two regions.

[6] 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 two-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 one's needs best. This approach was followed in our experiments.

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 using the 'early stopping' method, that is, stopping the training once the statistical performance of the model reaches a plateau or starts deteriorating.

The results obtained by Patuelli et al. (2006a, b) for different ex-post forecasting years show that statistical error is lower for the West German models (due to a longer time span of the dataset), and most importantly, that improvements in the accuracy of the forecasts are obtained when shift-share analysis components are implemented in the NN models (that is, in Models BSS, BSSN, BSSR). In addition, such models are found to outperform OLS and random walk models. However, 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. In this context, internal parameters and functions can play a critical role. The next section discusses the selection of appropriate NN parameters by means of a sensitivity analysis.

4 The role of the parameters: 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 focused 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 (Sect. 4.2) and of varying activation functions (Sect. 4.3), so to provide a more complete overview of NN setting issues. For our analysis, we use the baseline model presented in Sect. 3 (Model B), because of its simple application and stable performance seen in previous experiments. For each sub-analysis, we provide pooled MSE and MAPE obtained for the years 2001, 2002, 2003 and 2004. The computation of pooled error increases the reliability of our statistical findings, by averaging out the stochastic variability of the models' single-period application. In Sect. 5, we subsequently

evaluate the impact of the chosen set of parameters on the statistical performance of more complex NN models.

4.2 Learning rate and momentum

4.2.1 Description

The backpropagation algorithm (BPA) (see Sect. 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 can only assume positive values, between 0 and 1. Large values imply a quick learning of the network, while values that are too large may cause the NN to be unstable, therefore endangering 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.[7]

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.[8] Momentum values close to 1 will increase the influence of previous weights corrections on the current corrections, while an NN with a momentum close to 0 will mainly (or 'only', in the case of 0) rely, at each

[7] 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, to the discussion of adaptive LR to Sect. 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.

[8] 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 percent 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.

stage of the training, on the current correction.[9] 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 momentum, weight corrections in the NN training are channelled in the same direction of the preceding iteration.[10] Generally, experimenting with different values of momentum may be necessary, as for LR, in order 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 for each parameter: 0.1, 0.5 and 0.9. The resulting nine combinations can be separately tested, without excessive computation efforts, while covering most of the spectrum of possible values. The same approach is followed in our experiments, always using a sigmoid (logistic) activation function. 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 Sect. 2).

Table 1 shows the pooled MSE and MAPE obtained for the forecasting years from 2001 to 2004. 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.

We find that a low LR, matched with 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, 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 that performs significantly better than the rest.

[9] For example, a momentum value set at 0.5 means that 50 percent of the weight adjustment, at each stage, will be on the basis of the current error, while the remaining 50 percent 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.

[10] 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 that are more regular or smoother, or when the functional relationships to be learned are relatively simple.

Table 1 Sensitivity analysis for learning rate and momentum: Model B, West and East Germany, years 2001–2004

MSE (/1,000)				MAPE			
Learning rate	0.1	0.5	0.9	Learning rate	0.1	0.5	0.9
Momentum							
West Germany							
0.1	10,242.45 (6)	9,481.17 (3)	10,072.73 (5)	0.1	3.72 (4)	3.65 (2)	3.79 (6)
0.5	*9,226.85 (1)*	9,575.08 (4)	9,478.07 (2)	0.5	*3.59 (1)*	3.73 (5)	3.70 (3)
0.9	12,161.96 (9)	10,962.08 (7)	11,839.88 (8)	0.9	4.04 (8)	3.83 (7)	4.11 (9)
East Germany							
0.1	*2,391.72 (1)*	3,609.21 (8)	3,786.61 (9)	0.1	3.46 (4)	3.44 (3)	3.46 (5)
0.5	3,248.86 (6)	3,026.33 (4)	2,891.25 (2)	0.5	3.43 (2)	3.72 (7)	3.73 (8)
0.9	2,938.76 (3)	3,206.21 (5)	3,305.50 (7)	0.9	*3.42 (1)*	3.86 (9)	3.70 (6)

The ranking of the NN models is shown in brackets

This consideration stresses once again the crucial role played by the learning parameters in the performance of NNs.[11]

4.2.3 Adaptive learning rate

The BPA can suffer from 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 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 the adaptive learning rate (ALR) has been proposed. In the *bold driver* method (Vogl et al. 1988), the LR—as defined in Sect. 4.2.1—is augmented by a factor ρ when the error computed at iteration n is greater than at iteration $n - 1$. Otherwise, the LR is diminished by a factor σ when the error decreases.[12] 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 NN weight can have its own LR,

[11] The inconsistent results in the literature regarding the search for 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.

[12] 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. The momentum parameter can also be modified during learning: that is, it can be forced to 0 when the error increases and brought back to its value in the opposite case (Hagan et al. 1996).

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 weights. Moreover, in the *SuperSAB* method, it is suggested that the ρ and σ factors should be different in value, and that 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 cannot completely avoid the algorithm from stalling in slow convergence plateaus, since they use 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 fixed LR. We consider two NN models: the first one employs a LR of 0.1, while the second one uses an ALR. Both models have a momentum of 0.5, as found in Sect. 4.2.2. Again, a sigmoid activation function is used in both models. The ALR used is implemented as follows:

- The LR is modified at each training iteration. The extent of its recalculation is based on the error obtained 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 of training epochs (Fig. 1), the NNs with an ALR appear to reach a stable training error (converge) faster than the ones with fixed LR. This 'informal' result is consistent with the literature.

The subsequent question is whether 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

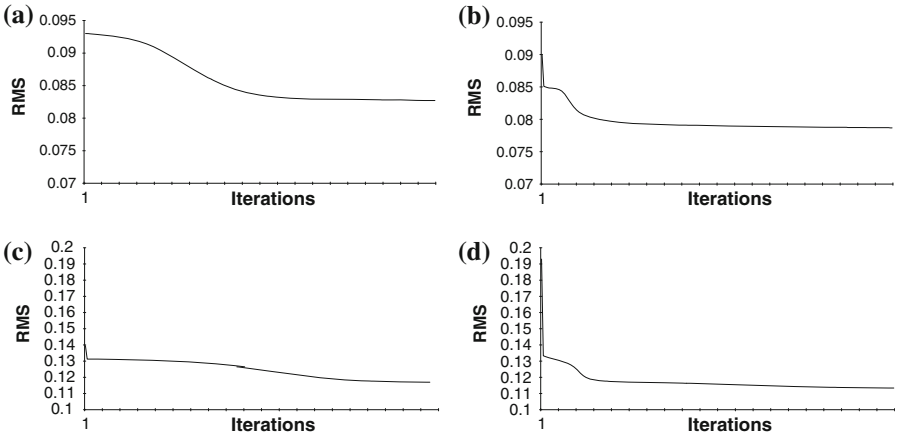

Fig. 1 Training error evolution over 400 iterations, for West and East German NN models: West Germany, fixed LR (**a**), ALR (**b**); East Germany, fixed LR (**c**), ALR (**d**)

Table 2 Sensitivity analysis for adaptive learning rate: Model B, West and East Germany, years 2001–2004

	West Germany		East Germany	
	MSE (/1,000)	MAPE	MSE (/1,000)	MAPE
Fixed LR (0.1)	*9,226.85* (*1*)	*3.59* (*1*)	3,248.86 (2)	*3.43* (*1*)
Adaptive LR	9,670.04 (2)	3.75 (2)	*3,229.53* (*1*)	3.45 (2)

The ranking of the NN models is shown between brackets

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 seen in the LR/momentum (above) and activation function (below) analyses.

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 that have a bigger error than Model 1 are expected to be 50 percent 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 percent of the cases. The ST statistic is computed as:

$$\mathrm{ST} = \left(C - \frac{N}{2}\right) \Big/ \left(\frac{1}{2}\sqrt{N}\right), \tag{2}$$

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 large samples, 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 4 years of simulated out-of-sample forecasts (2001–2004), obtaining 1,304 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 these analyses, we conclude that, in our experiments, ALR does not provide relevant approximation advantages in addition to a faster convergence of the algorithm. 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 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. However, the sigmoid function is 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 and hyperbolic functions are similar to the sigmoid, 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 ± 1. The Gaussian function forces small values to 1 and extreme values to 0. The augmented ratio function looks like an inverted Gaussian function. 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, when learning about deviations from the average is involved. Furthermore, a different activation function can ideally be used for each computational unit in the NN (for example, both linear and sigmoid functions, as in Wong 1991).[13]

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.[14] The activation functions

[13] 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. With regard to the latter, 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, provides no clear results on whether linear or nonlinear activation functions should be preferred for 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 aspect is not relevant in our case, as our NN models employ growth rates.

[14] The software used for our experiments does not allow selecting multiple simultaneous functions.

Table 3 Sensitivity analysis for activation functions: Model B, West and East Germany, years 2001–2004

West Germany	Sigmoid	Aug. ratio	Gaussian	Hyperbolic	Linear
MSE (/1,000)	*9,226.85 (1)*	9,297.49 (2)	10,131.27 (4)	9,945.25 (3)	12,307.48 (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 (/1,000)	3,248.86 (3)	3,678.93 (5)	2,653.34 (2)	3,315.57 (4)	*2,505.84 (1)*
MAPE	3.43 (3)	3.44 (4)	*3.41 (1)*	3.42 (2)	3.73 (5)

The ranking of the NN models is shown between brackets

tested here are: (1) sigmoid; (2) augmented ratio; (3) Gaussian; (4) hyperbolic; and (5) linear, as outlined above. While the linear function is normally used in the output layer only, our experiments test its implementation in the whole NN. All models employ the set of learning parameters (a LR of 0.1 and a momentum of 0.5) found in Sect. 4.1.2. Table 3 presents the results obtained for both West and East German models.

The statistical results shown in Table 3 generally confirm, in particular for the West German NN models, the results found in the literature: the 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 Sect. 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 trend.

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 may 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 specifications. Finally, the statistical results of the sensitivity analysis carried out above call for further testing, in particular in order 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. Such analysis is provided in the next section.

Table 4 Pooled statistical error of the NN models; West and East Germany, years 2001–2004

West	MSE (/1,000)	MAPE	East	MSE (/1,000)	MAPE
Model B	27,474.58 (3)	5.67 (3)	Model B	3,248.86 (2)	3.43 (5)
Model BD	25,983.19 (2)	5.10 (2)	Model BD	*2,543.62 (1)*	3.01 (2)
Model BSS	29,384.08 (4)	5.85 (4)	Model BSS	13,633.35 (6)	*2.86 (1)*
Model BSSN	41,228.08 (5)	7.18 (5)	Model BSSN	8,080.81 (3)	3.63 (6)
Model BSSR	55,694.54 (6)	7.78 (6)	Model BSSR	8,676.52 (5)	3.31 (4)
Model BW	*12,749.12 (1)*	*4.29 (1)*	Model BW	8,659.66 (4)	3.19 (3)

The ranking of the NN models is shown in brackets

5 Post-evaluation of different neural network model specifications

In the light of the findings of the sensitivity analysis carried out above, we evaluate the statistical performance of different NN model specifications exploiting the findings of Sect. 4. Table 4 presents the pooled statistical results computed for the six NN models presented in Sect. 3, on four forecasting periods: 2001, 2002, 2003 and 2004. The LR and momentum values used are 0.1 and 0.5, respectively, while a sigmoid activation function is employed.

The statistical results shown in Table 4 can be interpreted as follows:

- for West Germany:
  - 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
  - the shift-share-enhanced models (SS models) do not lead to better statistical performance;

- for East Germany, our results seem more unclear:
  - 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 sort out 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 the baseline model (Model B). The test statistic is -26.42, showing that Model BW, though 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 percent 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, when using the learning parameters and activation function chosen during the sensitivity analysis, the baseline model (Model B) and the district-type model (Model BD) emerge as the most suitable. However, with regard to the interpretation of these findings, attention should be focused on the use of socioeconomic covariates. The use of the wages and urbanization variables does not unequivocally improve the results, suggesting an 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.

On the one hand, this result confirms the problem of finding out which region-specific information is relevant for a specific case. On the other hand, the parameters chosen for our NN models might not be suitable for all model specifications, since they were tested on Model B only. They indeed appear to work for Model B (and a comparably simple models such as Model BD). But the new parameters appear to have a limited influence on the performance of NN models employing richer data (NN-SS models, employing SSA components).

It could then be argued that a specific class of NN models should first be selected, on which a specific sensitivity analysis concerning the parameters should be carried out.

6 Conclusions

In this paper we presented a analysis of the role of parameters with reference to the performance of NN models developed for regional employment forecasting in Germany. Our experiments can be divided in two phases. In the first phase, we carried out a sensitivity analysis, on a baseline model (Model B), in order to investigate the effects of varying learning parameters and functional forms on forecasting performance. In the second phase, in order to verify the suitability of the NN parameters set chosen on Model B, we tested five additional NN models: two models strongly related to Model B (BD and BW) and three incorporating shift-share analysis components (BSS, BSSN and BSSR), called NN-SS models.

Our analyses show that, for Model B, low learning rate (LR) values and medium momentum values improve the forecasts of our models. Moreover, we 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 this latter function may be deemed suitable for the case of East Germany (where the employment trends appear to be less complex). This result calls for testing on the linearity of the employment data, in particular for East Germany.

When testing a set of five additional NN models, we observed heterogeneous levels of statistical error, for both the West and East Germany models. In particular, we identified two preferred model specifications, viz. Model B for West Germany and Model BD for East Germany (introducing urbanization and agglomeration data). With regard to the class of NN-SS models, no comparable gain was obtained, most likely because of different levels of computational complexity and richness of information.

In summary, our results suggest that the choice of learning parameters *is* relevant, but cannot be generalized to NN models employing different inputs and structures. The reply to the question if NN parameters are more relevant than NN structures should then be sought in further detail, through a direct statistical comparison.

In the light of a further discussion of the NN parameters role, the paper can be expanded in different directions. From a methodological point of view, it may be desirable to test out more elaborate NN models, such as time-delay NNs (Waibel et al. 1989) or multi-function NNs. Also, a more in-depth analysis of the spatial interactions among districts might help to improve our understanding of regional phenomena. The incorporation, in Model BSSN, of information on the (employment) performance of the 'neighbours' is a first step in this direction. In future research, the potential of spatial statistics methods such as spatial filtering (Griffith 2003; Patuelli et al. 2010) for developing explicit spatial NN models should also be considered. In the same context, a spatial analysis of the NN residuals would certainly be helpful.

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, using new variables (such as unemployment or net migration) is also desirable.

References

Baltagi BH (2001) Econometric analysis of panel data, 2nd edn. Wiley, Chichester

Böltgen F, Irmen E (1997) Neue siedlungsstrukturelle regions-und kreistypen. Mitteilungen und Informationen der BfLR H 1:S. 4–S. 5

Chakraborty K, Mehrotra K, Mohan CK, Ranka S (1992) Forecasting the behavior of multivariate time series using neural networks. Neural Netw 5(6):961–970

Cooper JCB (1999) Artificial neural networks versus multivariate statistics: an application from economics. J Appl Stat 26:909–921

Cottrell M, Girard B, Girard Y, Mangeas M, Muller C (1995) Neural modelling for times series: a statistical stepwise method for weight elimination. IEEE Trans Neural Netw 51(2):240–254

Dunn ES (1960) A statistical and analytical technique for regional analysis. Pap Proc Reg Sci Assoc 6:97–112

Fischer MM (2001) Computational neural networks—Tools for spatial data analysis. In: Fischer MM, Leung Y (eds) Geocomputational modelling. Techniques and applications. Springer, Berlin, pp 15–34

Gopal S, Fischer MM (1996) Learning in single hidden layer feedforward neural network models: backpropagation in a spatial interaction modeling context. Geograph Anal 28(1):38–55

Gorr WL, Nagin D, Szczypula J (1994) Comparative study of artificial neural network and statistical models for predicting student grade point averages. Int J Forecast 10(1):17–34
Griffith DA (2003) Spatial autocorrelation and spatial filtering: gaining understanding through theory and scientific visualization. Springer, Berlin
Hagan MT, Demuth HB, Beale MH (1996) Neural network design. PWS Pub., Boston
Jacobs RA (1988) Increased rates of convergence through learning rate adaptation. Neural Netw 1(4):295–308
Klimasauskas CC (1991) Applying neural networks. part 3: training a neural network. PC/AI Magazine 5:20–24
Kuan CM, Hornik K (1991) Convergence of learning algorithms with constant learning rates. IEEE Trans Neural Netw 2:484–488
Kuan C-M, Liu T (1995) Forecasting exchange rates using feedforward and recurrent neural networks. J Appl Econom 10(4):347–364
Lehmann EL (1998) Nonparametrics: statistical methods based on ranks (rev. ed.). Prentice Hall, Upper Saddle River
Nazara S, Hewings GJD (2004) Spatial structure and taxonomy of decomposition in shift-share analysis. Growth Change 35(4):476–490
Park H, Amari S-I, Fukumizu K (2000) Adaptive natural gradient learning algorithms for various stochastic models. Neural Netw 13:755–764
Patuelli R, Reggiani A, Nijkamp P (2006a) The development of regional employment in Germany: results from neural network experiments. Sci Regionali 5(3):63–95
Patuelli R, Reggiani A, Nijkamp P, Blien U (2006b) New neural network methods for forecasting regional employment: an analysis of German labour markets. Spatial Econ Anal 1(1):7–30
Patuelli R, Longhi S, Reggiani A, Nijkamp P (2008) Forecasting regional employment in Germany by means of neural networks and genetic algorithms. Environ Plan B 35(4):701–722
Patuelli R, Griffith DA, Tiefelsdorf M, Nijkamp P (2010) Spatial filtering and eigenvector stability: space-time models for German unemployment data. Int Reg Sci Rev (forthcoming)
Plagianakos VP, Vrahatis MN and Magoulas GD (1999) Nonmonotone methods for backpropagation training with adaptive learning rate. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Washington, July, vol 3, pp 1762–1767
Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65:386–408
Rumelhart DE, McClelland JL (1986) Parallel distributed processing: explorations in the microstructure of cognition. MIT Press, Cambridge
Rumelhart DE, Durbin R, Golden R, Chauvin Y (1995) Backpropagation: the basic theory. In: Chauvin Y, Rumelhart DE (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 Comput Surv 27(4):519–542
Schintler LA, Olurotimi O (1998) Neural networks as adaptive logit models. In: Himanen V, Nijkamp P, Reggiani A (eds) Neural networks in transport applications. Aldershot Brookfield, Ashgate, pp 131–160
Sharda R, Patil RB (1992) Connectionist approach to time series prediction: an empirical test. J Intell Manuf 3(5):317–323
Srinivasan D, Liew AC, Chang CS (1994) A neural network short-term load forecaster. Elect Power Syst Res 28:227–234
Tang Z, Fishwick PA (1993) Feedforward neural nets as models for time series forecasting. INFORMS J Comput 5(4):374–385
Tang Z, Almeida C, Fishwick PA (1991) Time series forecasting using neural networks vs Box-Jenkins methodology. Simulation 57(5):303–310
Tollenaere T (1990) SuperSAB: fast adaptive back propagation with good scaling properties. Neural Netw 3(5):561–573
Vogl TP, Mangis JW, Rigler AK, Zink WT, Alkon DL (1988) Accelerating the convergence of the back-propagation method. Biol Cybern 59:257–263
Waibel AH, Hanazawa T, Hinton GE, Shikano K, Lang KJ (1989) Phoneme recognition using time-delay neural networks. IEEE Trans Acoust Speech Signal Process 37(3):328–339
Weisstein EW (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 FS (1991) Time series forecasting using backpropagation neural networks. Neurocomputing 2(4):147–159

Yu XH, Chen GA, Cheng SX (1995) Dynamic learning rate optimization of the backpropagation algorithm. IEEE Trans Neural Netw 6(3):669–677

Zhang G, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks: the state of the art. Int J Forecast 14(1):35–62