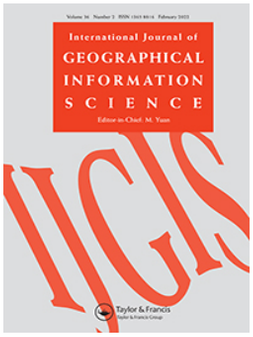

International Journal of Geographical Information Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

A geographically weighted artificial neural network

Julian Hagenauer & Marco Helbich

To cite this article: Julian Hagenauer & Marco Helbich (2022) A geographically weighted artificial neural network, International Journal of Geographical Information Science, 36:2, 215-235, DOI: 10.1080/13658816.2021.1871618

To link to this article: https://doi.org/10.1080/13658816.2021.1871618

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
2022, VOL. 36, NO. 2, 215–235
https://doi.org/10.1080/13658816.2021.1871618

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

OPEN ACCESS Check for updates

A geographically weighted artificial neural network

Julian Hagenauer and Marco Helbich

Department of Human Geography and Spatial Planning, Faculty of Geosciences, Utrecht University, Utrecht, The Netherlands

ABSTRACT

While recent developments have extended geographically weighted regression (GWR) in many directions, it is usually assumed that the relationships between the dependent and the independent variables are linear. In practice, however, it is often the case that variables are nonlinearly associated. To address this issue, we propose a geographically weighted artificial neural network (GWANN). GWANN combines geographical weighting with artificial neural networks, which are able to learn complex nonlinear relationships in a data-driven manner without assumptions. Using synthetic data with known spatial characteristics and a real-world case study, we compared GWANN with GWR. While the results for the synthetic data show that GWANN performs better than GWR when the relationships within the data are nonlinear and their spatial variance is high, the results based on the real-world data demonstrate that the performance of GWANN can also be superior in a practical setting.

ARTICLE HISTORY
Received 10 April 2020
Accepted 31 December 2020

KEYWORDS
Geographically weighted regression; artificial neural network; spatial heterogeneity; nonlinear relationships; spatial prediction

1. Introduction

Spatial heterogeneity of relationships (i.e. spatial nonstationarity) is an important issue in spatial data analysis (Anselin 1989). It refers to the notion that for a spatial process, the relationships between variables depend to some degree on the location where the relationships are observed (Fotheringham *et al*. 2002). If spatial heterogeneity is not appropriately taken into account when calibrating a model, the estimation of the coefficients is likely to be biased, which can lead to inappropriate conclusions (Páez *et al*. 2008, LeSage and Pace 2009).

Several approaches have been proposed to model spatially varying relationships. Notable examples include the expansion method (Casetti 1972), weighted spatial adaptive filtering (Gorr and Olligschlaeger 1994), Eigenvector spatial filtering (Griffith 2003), and geographically weighted regression (GWR) (Brunsdon *et al*. 1999). Of these approaches, GWR has received the most attention and is employed across many disciplines, for example, real estate economics (Bitter *et al*. 2007, Helbich and Griffith 2016), ecology (Nelson *et al*. 2007), criminology (Waller *et al*. 2007, Troy *et al*. 2012), health (Choi and Kim 2017), and land-use science (Yu *et al*. 2011, Hagenauer and Helbich 2018).

CONTACT Julian Hagenauer julian.hagenauer@gmx.de
Supplemental data for this article can be accessed here.

GWR is an extension of ordinary least squares (OLS), which estimates for each location a weighted least squares regression, where observations that are closer to the regression location are given a higher weight than those farther away. The weighting is determined by a distance–decay kernel function and a bandwidth parameter.

Several extensions and modifications of GWR have been proposed. While in basic GWR all relationships are assumed to vary spatially, in mixed GWR (Brunsdon *et al*. 1999) only a subset of the coefficients are subject to geographical weighting; the kernel function and bandwidth for each spatially varying coefficient are identical. The latter restriction was addressed by Fotheringham *et al*. (2017), who proposed a multiscale GWR that uses individual bandwidths for the coefficients to model different scales of spatial heterogeneity. Furthermore, while basic GWR is based on Euclidean distances between observations, the application of different distance metrics has been proposed. Lu *et al*. (2011), for example, showed that non-Euclidean distance metrics can improve the fit of GWR, whereas Fotheringham *et al*. (2015) suggested the use of a spatio-temporal distance metric. Lu *et al*. (2017) combined multiscale GWR with individual distance metrics per coefficient.

GWR has also been criticized for artificially introducing multicollinearity between coefficient pairs (Wheeler and Tiefelsdorf 2005), which was recently refuted (Fotheringham and Oshan 2016). To counteract this criticism, penalized forms of GWR were proposed (e.g. geographically weighted lasso (Wheeler 2009), ridge (Wheeler 2009), ridge (Wheeler 2007, Bárcena *et al*. 2014), and elastic net regression (Li and Lam 2018)). Another extension is geographically neural network weighted regression (Du *et al*. 2020), which utilizes an artificial neural network (ANN) to find appropriate geographical weights when estimating the coefficients of a GWR model.

Despite these efforts, some restrictions of GWR have not yet been addressed. For instance, because GWR resembles a collection of local models where data from neighboring local models are reused, its inferential properties are inferior to a single nonstationary model (Comber *et al*. 2020). Also, analogous to OLS, when using GWR in its simplest form it is assumed that the relationships between dependent and independent variables are linear. This assumption, however, typically does not hold for complex spatial prediction tasks (Anselin 1989, Leuenberger and Kanevski 2015).

To address this issue, we propose a geographically weighted artificial neural network (GWANN), which combines geographical weighting with an ANN. Similar to GWR, GWANN uses a distance-decay kernel function and a bandwidth parameter to geographically weight observations when building the model. However, in contrast to GWR, GWANN is also able to model nonlinear functions in a data-driven manner without making any assumptions.

The rest of this article is structured as follows. Section 2 describes GWR and ANN and introduces GWANN. Next, section 3 presents experiments that were carried out to compare GWANN with GWR. Finally, section 4 gives concluding remarks and proposes future work.

2. Methods

2.1. Artificial neural network

An artificial neural network (ANN) consists of a set of neurons and unidirectional connections between them, which enables the imitation of the brain's ability to detect patterns and learn relationships within data (Haykin 2008). Associated with each neuron i is an activation function ϕ_i and each connection between two neurons i,j has a weight w_{ij} assigned that controls the influence of neuron i on neuron j. While the neurons represent the basic computation units of an ANN, the weighted connections between them allow the modeling of complex relationships.

The neurons are typically organized in layers, and each neuron in a layer has directed connections to the neurons in the subsequent layer (Figure 1). The first layer is termed 'input layer' and the last layer 'output layer,' while all layers in between are 'hidden layers'. The input data are passed from the input layer to the first hidden layer, where it is aggregated and transformed as follows:

$$net_j = \sum_{i \in P_j} w_{ij} o_i \tag{1}$$

where w_{ij} is the weight of the connection between neuron i and j, o_i the output of neuron i, and P_j the set of neurons that have an outgoing connection to neuron j. The output of a neuron i is calculated as follows:

$$o_i = \phi(net_i) \tag{2}$$

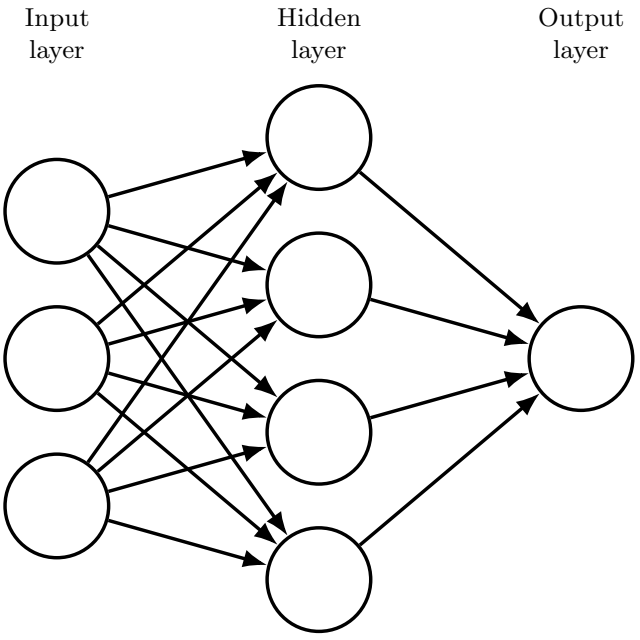

Figure 1. ANN with three layers.

where ϕ is the activation function of neuron i. A common activation function is the hyperbolic tangent function, which is defined as $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. This function is particularly useful because it is continuous and differentiable; both are necessary conditions for calculating the network's error gradient (Rojas 2013).

The output of each neuron is then passed on to the neurons in the next layer. For each subsequent layer, this procedure is repeated until the output layer of the network is reached. The output of the output layer represents the total output of the network.

In order to model nonlinear relationships, the connection weights of an ANN must be adjusted. This is typically done using a two-step procedure. In the first step, the error signal of each neuron for a given observation is calculated using backpropagation (Rumelhart *et al*. 1986). The error signal depends on the error function. In the case of regression, the error function is defined as $E = \frac{1}{2}\sum_{i=1}^{n}(t_i - o_i)^2$ where t_i is the target value, o_i the output of the output neuron i, and n the number of the target values. Given this error function, the error signal is calculated as follows:

$$\delta_j = \begin{cases} \phi'(net_j)(o_j - t_j) & \text{if } j \text{ is an output neuron} \\ \phi'(net_j)\sum_k \delta_k w_{jk} & \text{otherwise} \end{cases} \tag{3}$$

where o_j is the output of neuron j, t_j the target value of neuron j, w_{jk} the connection weight between neuron j and k, δ_k the error signal for neuron k, net_j the network input to neuron j, and ϕ' the derivative of the activation function.

In the second step, the connection weights are adjusted using gradient descent:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \delta_j o_i \tag{4}$$

where w_{ij} is the connection weight between neuron i and j, E the error function, o_i the output of neuron i, and δ_j the error signal of neuron j. Both steps are repeated until a terminating condition is reached (e.g. the error rate is below a predetermined threshold value).

Several extensions and variants of gradient descent have been proposed to improve the training of the network. To make the training more robust to noise, the error gradients are in practice summed over a subset of observations, termed a 'mini-batch'. The connection weights are then updated using the accumulated changes. Also, using Nesterov's accelerated gradient (Nesterov 1983) when adjusting the connection weights can substantially improve the training performance (Sutskever *et al*. 2013).

2.2. Geographically weighted regression

Geographically weighted regression (GWR) (Brunsdon *et al*. 1996) estimates for each location a separate local model. Assuming that there are n locations and each location has an observation assigned to it, the GWR model for the location $i \in 1, 2, \ldots, n$ is:

$$y_i = \sum_{j=0}^{m} \beta_{ij} x_{ij} + \in_i \tag{5}$$

where y_i is the dependent variable, x_{ij} the independent variable j, β_{ij} the coefficient for the independent variable j, and $\in_i$ the error term, which is assumed to be independent and identically distributed.

GWR weights the observations by their spatial distance when estimating the local coefficients; close observations are given more weight than observations farther away. The estimation is typically done using weighted least squares, the matrix expression of which is:

$$\hat{\beta}_i = (X^T W_i X)^{-1} X^T W_i y \tag{6}$$

where X is the design matrix, y the dependent variable, and W_i a column vector of the spatial weights matrix W for location i. To calculate W, a kernel function is applied to the distances between observations and regression locations. Widely used kernels are Gaussian, bisquare, tricube, and boxcar kernels (Brunsdon *et al*. 1999). The Gaussian kernel, for instance, is defined as $v_{ij} = e^{-0.5(\frac{d_{ij}}{h})^2}$, where d_{ij} is the distance between locations i and j and h is the kernel bandwidth. The bandwidth determines the degree of variation in the local coefficient estimates and is considered to be more important for the performance of GWR than the choice of the kernel function (Fotheringham *et al*. 2002). The bandwidth can be either fixed or adaptive, where the latter refers to the distance to the k-nearest neighbor of each observation (Brunsdon *et al*. 2007, Guo *et al*. 2008).

2.3. Geographically weighted artificial neural network

A geographically weighted artificial neural network (GWANN) is a variant of an ANN that incorporates geographical weighting of connection weights. The principle idea is as follows. A basic ANN consists of an input, a hidden, and an output layer. The connection weights of a basic ANN from the hidden to the output layer can be interpreted as the coefficients of a linear model of nonlinearly transformed variables, namely the outputs of the hidden neurons. Thus, when the connection weights between the hidden and the output layer are estimated by utilizing a geographically weighted error function, these weights can be interpreted as a GWR model.

The architecture of GWANN is identical to that of a basic ANN, except that each output neuron of GWANN is assigned to a location in geographic space (Figure 2). This allows to calculate the spatial distances between the observations and the locations of the output neurons.

Besides the network architecture, the main difference between GWANN and a basic ANN is that GWANN uses a geographic weighted error function instead of the basic quadratic error function in order to calculate an error signal. In the case of regression, the geographically weighted error function is defined as $E = \frac{1}{2}\sum_{i=1}^{n} v_i(t_i - o_i)^2$, where t_i is the target value, o_i the output of output neuron i, v_i the geographically weighted distance between the observation and the location of output neuron i, and n the number of target values/output neurons. Following this definition, the difference between the output neurons' output and the target values is weighted by the spatial distance between output neurons' location and the observation; when the output neurons' location and observation are close, the difference is given more weight than when they are farther apart. Note that the number of target values must be identical to the number of output neurons. In

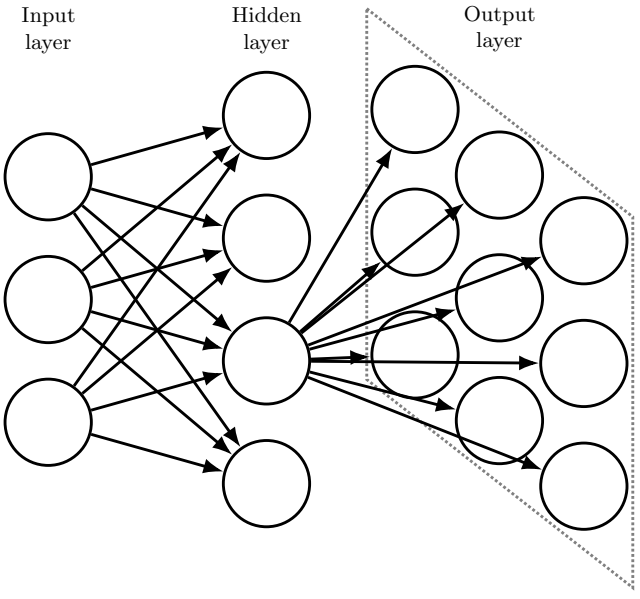

Figure 2. GWANN with three layers. The rectangle indicates that the output neurons are assigned to locations on a plane. Note that although each hidden neuron typically has connections to all output neurons, for the sake of clarity the outgoing connections are shown for a single hidden neuron only.

particular, in a practical example where one wants to calculate the value of the geographic error function for a single target value but multiple output neurons with typically different locations, it is necessary to replicate the target value for each output neuron.

Following the definition of the geographically weighted error function, the calculation of the error signal of backpropagation is modified as follows:

$$\delta_j = \begin{pmatrix} \phi'(net_j) v_j (o_j - t_j) & \text{if } j \text{ is an output neuron} \\ \phi'(net_j) \sum_k \delta_k w_{jk} & \text{otherwise} \end{pmatrix} \tag{7}$$

where o_j is the output of neuron j, t_j the target value of neuron j, w_{jk} the connection weight between neuron j and k, δ_k the error signal for neuron k, net_j the network input to neuron j, ϕ' the derivative of the activation function, and v_j the geographically weighted distance between the observation and the location of output neuron j. Geographical weighting is only used for calculating the error signal of the output neurons, whereas all other neurons backpropagate the error signal of the neurons of the next layer. Like ANN, the connection weights of GWANN are adjusted using gradient descent (Equation (4)).

3. Experiments

To compare GWR with GWANN, we used four synthetic datasets and one real-world dataset from real-estate economics. The synthetic datasets gave us full control over the characteristics of the data, in particular the nature of the relationships and spatial heterogeneity, which contributed to a better understanding of the different properties of the models. The real-world data allowed us to assess the models in a practical use case.[1]

For all experiments, we scaled the input variables to have zero mean and unit variance to make them comparable. We used Nesterov's accelerated gradient with a momentum coefficient of 0.900 when adjusting the connection weights. We set the learning rate η of GWANN to 0.010 and the mini-batch size to 50. While in principle the number of hidden layers of an ANN is arbitrary, we chose networks with a single hidden layer. Given enough hidden neurons, ANNs with a single hidden layer are able to arbitrary well approximate any continuous function on closed and bounded subsets of n-dimensional Euclidean space (Cybenko 1989). For each experiment, we tested different numbers of hidden neurons. A bias neuron is always added to the input and the hidden layer, but we did not include them when reporting the number of neurons. The hyperbolic tangent function is used as activation function for the hidden neurons.

We used a Gaussian kernel with GWR and GWANN for geographical weighting. When using an adaptive bandwidth, a grid search is performed to determine an appropriate bandwidth. When using a fixed bandwidth, the following local search approach is used to determine an appropriate bandwidth. The approach initially selects half of the largest distance between two observations as the current bandwidth. Then, a grid search is performed within the neighborhood of the current bandwidth for a bandwidth that results in a better mean performance. When one is found, the process is repeated within a smaller neighborhood of the newly found bandwidth until convergence.

The performance within the bandwidth search is estimated using 10-fold cross-validation (CV). This procedure randomly partitions the data into 10 disjoint subsets. One subset at a time is then used to test the model, while the others are used to build it. Then, the mean performance over all folds is reported. We used the root mean square error (RMSE) as a performance measure.

The number of training iterations of GWANN was also determined using 10-fold CV. Within each fold, the models are trained until the performance for the test data of the current fold does not improve for 1,000 iterations. The purpose of the additional iterations is to give the networks a chance to escape from local minima. This approach, commonly termed 'early stopping with patience', substantially reduces the risk of overfitting the training data (Bengio 2012). Then, the iteration for which the best mean performance over all folds has been obtained as well as the obtained performance value are reported.

3.1. Experiment 1: synthetic data

The purpose of this experiment was to investigate the differences between GWR and GWANN when modeling processes with different spatial characteristics. In particular, we were interested in how the model performance depends on the linearity and spatial variation of the relationships. We also examined the visualization of GWANN's connection weights between the hidden and output neurons as surfaces.

3.1.1. Data generating process

We created four artificial datasets. The spatial layout of the datasets was given by a grid of size 25×25. The following functions were used to create the datasets:

$$y_i = \beta_0 + \beta_1(u_i, v_i)x_{i1} + \beta_2^1(u_i, v_i)x_{i2} + \in_i \tag{8}$$

$$y_i = \beta_0 + \beta_1(u_i, v_i)x_{i1} + \beta_2^2(u_i, v_i)x_{i2} + \in_i \tag{9}$$

$$y_i = \beta_0 + 4\tanh(\frac{\beta_1(u_i, v_i)x_{i1}}{3}) + 4\tanh(\frac{\beta_2^1(u_i, v_i)x_{i2}}{3}) + \in_i \tag{10}$$

$$y_i = \beta_0 + 4\tanh(\frac{\beta_1(u_i, v_i)x_{i1}}{3}) + 4\tanh(\frac{\beta_2^2(u_i, v_i)x_{i2}}{3}) + \in_i \tag{11}$$

For all functions, (u_i, v_i) denotes the position of grid cell i, $\in_i$ the error term drawn from $N(0, 0.25)$, x_i a random variable drawn $N(0, 1)$, and β_0, $\beta_1(u_i, v_i)$, and $\beta_2^1(u_i, v_i)$ and $\beta_2^2(u_i, v_i)$, respectively, the coefficients for grid cell i. While the first two functions (Equations (8) and (9)) model linear relationships between the dependent and independent variables, the third and fourth functions (Equations (10) and (11)) use the hyperbolic tangent function to represent nonlinear relationships.

The coefficients were designed to represent different characteristics of spatial heterogeneity. They were calculated as follows:

$$\beta_0 = 1 \tag{12}$$

$$\beta_1(u_i, v_i) = 1 + \frac{u_i + v_i}{12} \tag{13}$$

$$\beta_2^1(u_i, v_i) = 1 + 2(\cos(\frac{\pi u_i}{24})\cos(\frac{\pi v_i}{24})) \tag{14}$$

$$\beta_2^2(u_i, v_i) = 1 + 2(\cos(\frac{\pi u_i}{12})\cos(\frac{\pi v_i}{12})) \tag{15}$$

For all coefficients, (u_i, v_i) denotes the position of grid cell i. β_0 represents a constant surface with no spatial heterogeneity. β_1 is a linear trend surface. β_2^1 and β_2^2 vary nonlinearly with location; the spatial variation of β_2^2 is higher than that of β_2^1. In terms of scale, β_2^1 represents small-scale spatial heterogeneity, and β_2^2 large-scale spatial heterogeneity. Figure 3 shows the coefficients' surfaces.

Following the definition of the coefficients, the first and third functions (Equations (8) and (10)) represent processes with low spatial variance, while the second and fourth functions (Equations (9) and (11)) represent processes with high spatial variance of coefficients.

3.1.2. Experimental setup

For all datasets, we used the variable y as the dependent variable and the variables x_1 and x_2 as independent variables. We used fixed bandwidths for GWR and GWANN. This allowed a finer control of the bandwidths when the observations were uniformly arranged in a grid and thus only a few distance classes were present.

To investigate the performance of GWR and GWANN, we used 10-fold CV with 90% of the data to determine an appropriate bandwidth for GWR and GWANN as well as an appropriate number of iterations for GWANN. Then, we used the same data to build a GWR and GWANN model with the hyperparameters determined and used the remaining data to obtain an

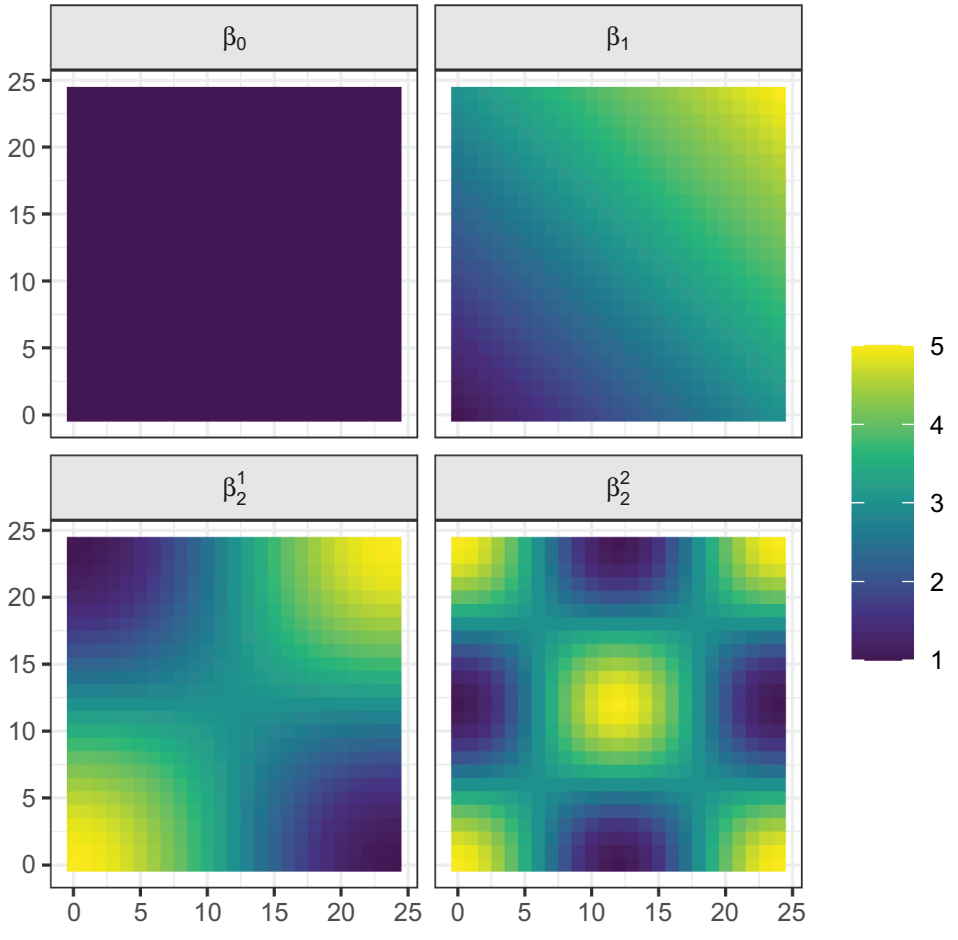

Figure 3. Coefficients' surfaces with different characteristics of spatial heterogeneity.

independent estimate of their performance. We repeated the procedure for 100 random replications of each of the four toy datasets and reported the mean results.

The estimated coefficients of GWR can be visualized as surfaces to explore the spatial variation of the relationships. Analogously to GWR, it is also possible to visualize GWANN's connection weights between the hidden and the output neurons as surfaces. Each surface then refers to a hidden neuron's output, which is a nonlinearly transformed linear combination of the input variables.

To investigate and compare the visualization of the coefficient surfaces of GWR and the connection weight surfaces of GWANN, we built a GWR and a GWANN model using an exemplary replication of the dataset that was created using Equation (11). This dataset is the most complex one because of the nonlinearity of the relationships and the high spatial variance of the coefficients. The number of hidden neurons of GWANN was set to five because this allowed a comprehensive visualization while providing a good model fit. Since we wanted to visualize the coefficient weights for every observation, the number of output neurons equaled the total number of observations, and each output neuron was assigned the location of an observation. Due to randomness in the data generating process and in the training of

GWANN, a different bandwidth and different number of training iterations were determined for most replications. We chose the bandwidth and number of iterations corresponding to the replication for which the median RMSE over all replications had been obtained.

3.1.3. Results & discussion

Figure 4 shows the mean number of training iterations GWANN until convergence. The mean number of training iterations of GWANN does not change with the number of hidden neurons when the relationships are nonlinear and the spatial variance of the coefficients is low; otherwise, it decreases with the number of hidden neurons.

Figure 5 shows the obtained mean bandwidths. The mean bandwidth of GWANN always decreases with the number of hidden neurons; the decrease, however, is small when the relationships are nonlinear. The mean bandwidth of GWANN is larger than that of GWR when the relationships are linear, whereas it is lower when the relationships are nonlinear. Also, the mean bandwidths of GWANN and GWR are generally higher when the spatial variance of the coefficients is low.

Figure 6 shows the mean RMSE of the models for the independent hold-out test datasets (for the proportion of explained variance, see Figure S1 in the supplementary materials). The mean RMSE of GWR is lower than the mean RMSE of GWANN when the relationships are linear. However, when the relationships are nonlinear, the mean RMSE of GWR is substantially higher than the mean RMSE of GWANN. This is not unexpected, because unlike GWANN, GWR is not inherently capable of modeling nonlinear relationships. The mean RMSE of GWANN is substantially lower than the mean RMSE of GWR when the relationships are nonlinear and the spatial variance of the coefficient is high. The mean RMSE of GWANN generally decreases with the number of hidden neurons when

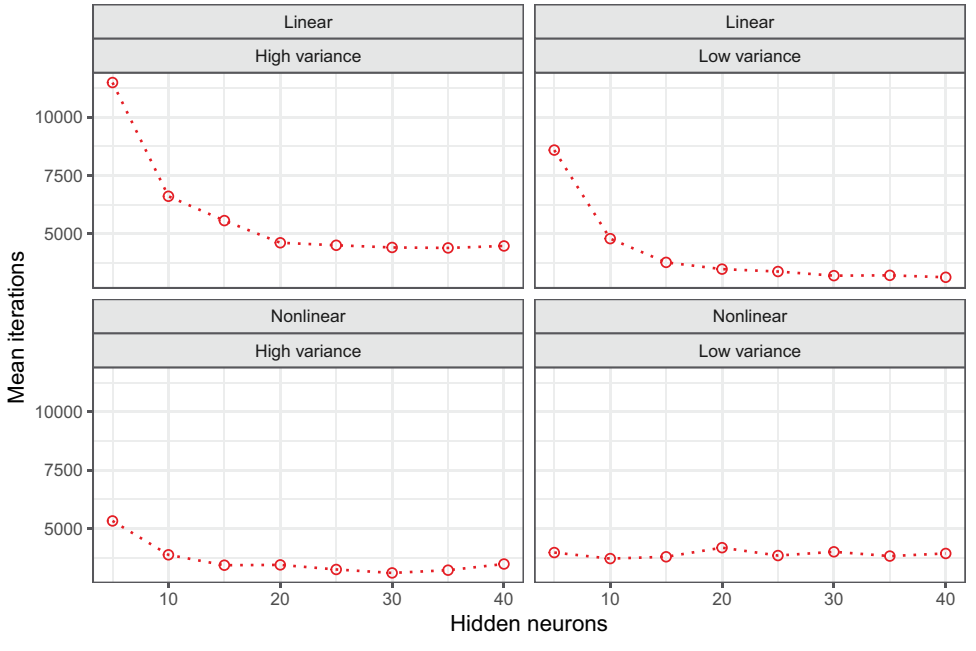

Figure 4. Number of iterations until convergence.

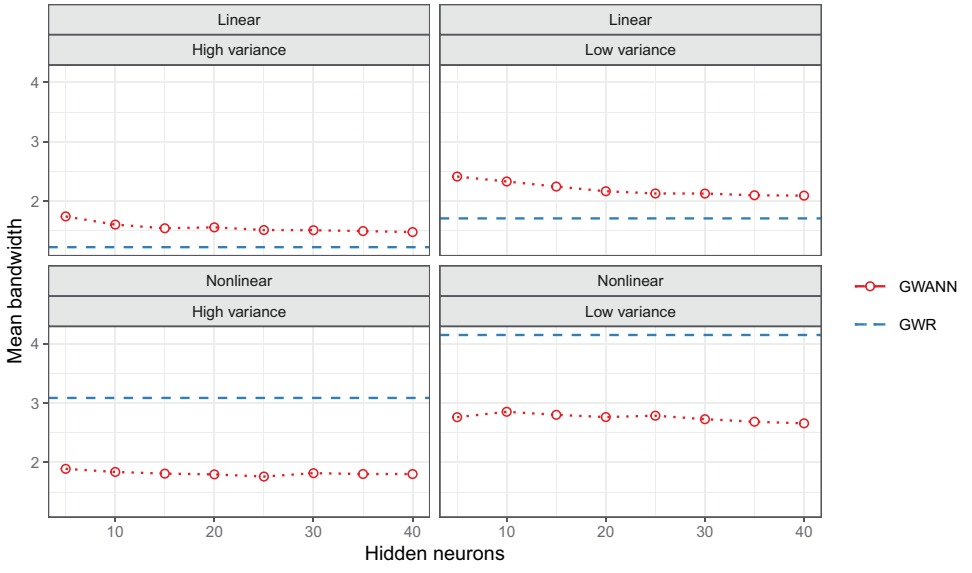

Figure 5. Determined bandwidths (fixed).

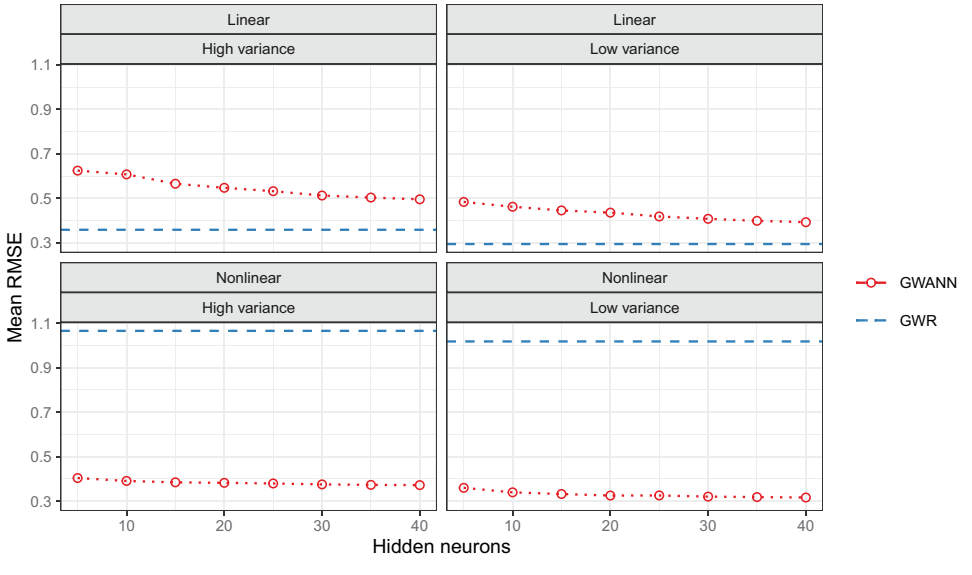

Figure 6. Estimated prediction error (RMSE).

the relationships are linear; the decrease is stronger, however, when the spatial variance of the coefficients is high. When the relationships are nonlinear and the spatial variance of the coefficients is low, then the mean RMSE of GWANN only decrease with the number of hidden neurons if that number is low; otherwise, the RMSE remains the same. No correlation between the number hidden neurons and mean RMSE is observable when the relationships are nonlinear and the spatial variance of the coefficients is high.

In general, the performance of all tested models depends on the nature of the underlying process. When the relationships in the data are nonlinear and the spatial variance of the coefficients is high (i.e. large-scale spatial heterogeneity), GWANN performs substantially better than GWR. In practice, however, the characteristics of the data generating process are usually not known beforehand and therefore it is necessary to empirically assess the performance of the competing models.

Using an exemplary replication of the dataset that was created using Equation (11), we trained GWANN for $5,610$ iterations with a bandwidth of 1.801 and fitted a GWR model with a bandwidth of 2.000. Figure 7 shows the coefficient surfaces of GWR. The coefficient surfaces roughly resemble the coefficients of the original dataset (see Figure 3). We calculated Pearson's correlation coefficient between the surfaces of GWR and the coefficients of the dataset to quantify their similarity. The linear trend of β_1 from bottom left to top right is observable for the surface of $\hat{\beta}^1$ ($r = 0.871$, $p \leq 0.05$) as well as the hill and valley patterns of β_2^2 for the surface of $\hat{\beta}_2^2$ ($r = 0.708$, $p \leq 0.05$). However, all surfaces of the estimated coefficients show irregularities and noise.

Figure 8 shows the connection weights between the hidden neurons (including the bias neuron) and the output neurons of GWANN as surfaces. Some surfaces of GWANN show patterns that correspond to the coefficients of the original dataset. The linear trend of β_1 from bottom left to top right is visible for the surfaces of neurons 2 (Pearson's correlation coefficient $r = -0.906$, $p \leq 0.05$) and 5 ($r = -0.779$, $p \leq 0.05$), whereas the hill and valley patterns of β_2^2 are noticeable for the surfaces of neuron 4 ($r = 0.960$, $p \leq 0.05$). The surfaces of neuron 1 and 3 and the bias neuron, however, do not resemble any of the coefficient surfaces. Also, none of the neurons' surfaces shows the pattern of β_0 and all surfaces of GWANN exhibit substantial traces of irregularities and noise. With the exception of β_0, we can identify for each coefficient at least one surface of GWANN's connection weights with which it is more correlated than it is with any surface of GWR's estimated coefficient.

While for this experiment the visualization of GWANN's surfaces provided evidence that the model learned the spatial relationships of the dataset, a detailed interpretation of the surfaces is difficult due to the complexity of the computations performed within the network. For instance, the surfaces of neurons 1 and 3 do not reveal how the neurons relate to the input data or how they contribute to the overall output of the network.

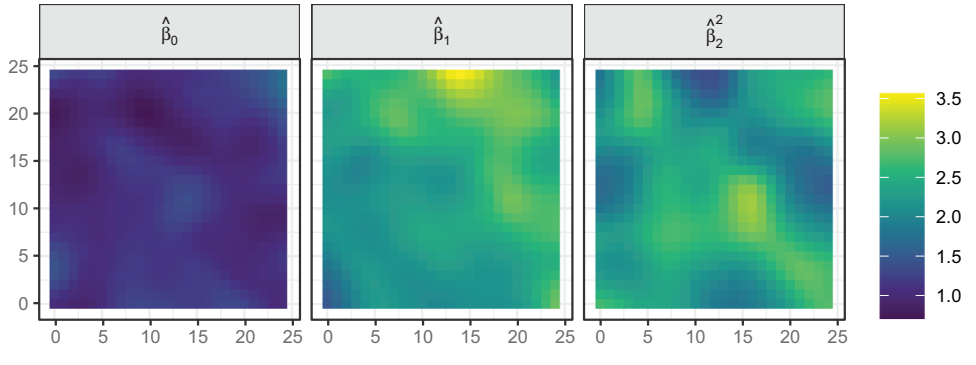

Figure 7. Estimated coefficient surfaces of GWR.

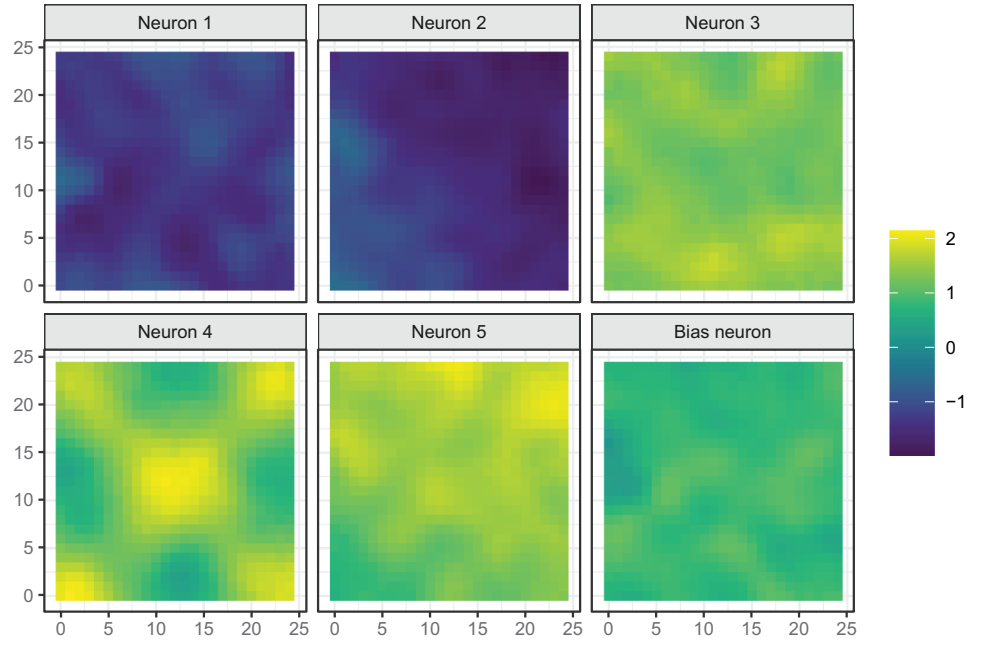

Figure 8. Connection weights between the hidden neurons (including the bias neuron) and the output neurons of GWANN as surfaces.

Moreover, the computations performed within the network become less traceable the more input and hidden neurons the network consists of, which further limits the usefulness of GWANN's surfaces for explorative spatial data analysis in a practical setting.

3.2. Experiment 2: house prices in Austria

In this experiment, we assessed the differences in the predictive performance of GWR and GWANN using real-world data. We also investigated the effect of different distance matrices on the predictions of GWR and GWANN and evaluated the spatial distribution of the residuals.

We chose housing as a case study because in real-estate economics, regression-based house price assessments are vital (Sopranzetti 2010). Hedonic theory assumes that a property represents a heterogeneous good that can be decomposed into its utility-bearing characteristics, and that the resulting benefit is reflected in the property price (Rosen 1974). Both the physical characteristics of a property (e.g. floor area) and the neighborhood characteristics (i.e. a dwelling's surroundings) contribute to the overall price. It is well established in housing research that transaction prices vary spatially and thus hedonic house price models that consider spatial heterogeneity are increasingly applied (e.g. Bitter *et al*. 2007, Sunding and Swoboda 2010, Lu *et al*. 2011, Helbich and Griffith 2016).

3.2.1. Data

Data on 3,887 geocoded single-family houses in Austria were provided by UniCredit Bank Austria AG (Helbich *et al*. 2014). Individual transaction prices of house purchases recorded

in euros were collected from 1998 to 2009, along with 11 structural properties of the houses and two temporal variables. Descriptive statistics are listed in Table S1 in the supplementary materials.

3.2.2. Experimental setup

We used the log-transformed transaction prices as the dependent variable and the structural properties and temporal variables as independent variables. We used an adaptive bandwidth for GWANN and GWR, because of the uneven distribution of the housing locations (Figure 12). We applied two different distance metrics for geographical weighting, namely Euclidean distance (ED) and travel time distance by car (TTD). TTDs were computed using the Open Source Routing Machine (Huber and Rust 2016) with OpenStreetMap data.

To investigate the performance of GWR and GWANN, we used 10-fold CV to obtain robust estimates of the performance of the models (outer 10-fold CV). Note that within each fold, 10-fold CV was also used to determine an appropriate bandwidth for GWR and GWANN and number of iterations for GWANN (inner 10-fold CV).

To investigate the predictions and residuals in detail, we built the models using the complete dataset. We chose the number of the networks' hidden neurons according to the number of hidden neurons for which the lowest mean RMSE had been obtained. Since we wanted to predict house prices for the complete dataset, the number of output neurons of GWANN equaled the total number of observations and each output neuron was assigned the location of an observation. Due to randomness in the (outer) 10-fold CV procedure and in the training of GWANN, a different bandwidth and a different number of training iterations were determined for most folds. We chose the bandwidth and number of iterations corresponding to the fold for which the median RMSE over all folds had been obtained.

If a model is unable to take into account the spatial properties of the data, its residuals tend to be spatially autocorrelated. We tested for residual spatial autocorrelation of the models using Moran's *I*. We calculated the test statistics using inverse EDs and evaluated the significance by means of 999 Monte-Carlo simulation runs.

3.2.3. Results & discussion

Figure 9 shows the mean number of training iterations of GWANN until convergence. GWANN generally requires fewer iterations to converge when using TTDs rather than EDs. The mean number of training iterations of GWANN when using EDs and TTDs decreases with the number of hidden neurons; the larger the number of hidden neurons, though, the smaller the decrease.

Figure 10 shows the obtained mean bandwidths for GWANN and GWR. While the mean bandwidth of GWANN is independent of the number of hidden neurons, the mean bandwidth is smaller when it uses TTDs rather than EDs. Similarly, the mean bandwidth of GWR is smaller when TTDs are used rather than EDs. Generally, the mean bandwidth of GWANN is considerably smaller than that of GWR, independent of the used distance metric and the number of hidden neurons. This result suggests that GWANN is generally able to model spatial variations in the data on a smaller scale than GWR.

Figure 11 shows the models' mean RMSEs obtained by means of (outer) 10-fold CV (for the proportion of explained variance, see Figure S2). While the mean RMSE of GWANN is lower when using EDs rather than TTDs, the difference in mean RMSE between the

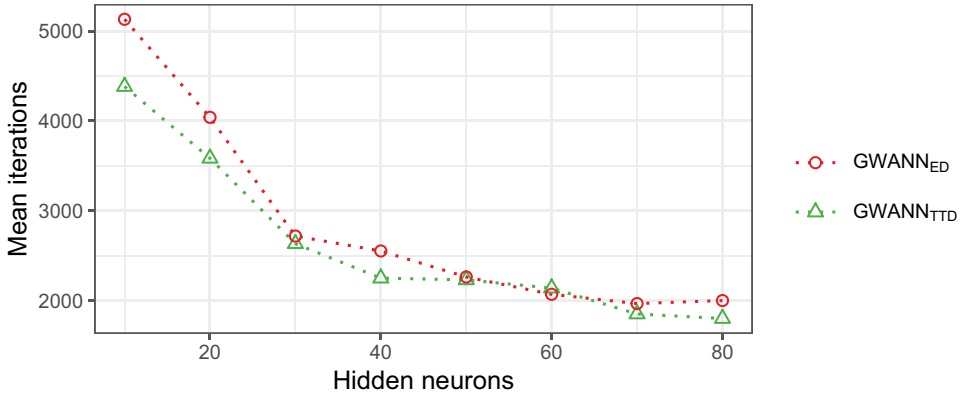

Figure 9. Number of iterations until convergence.

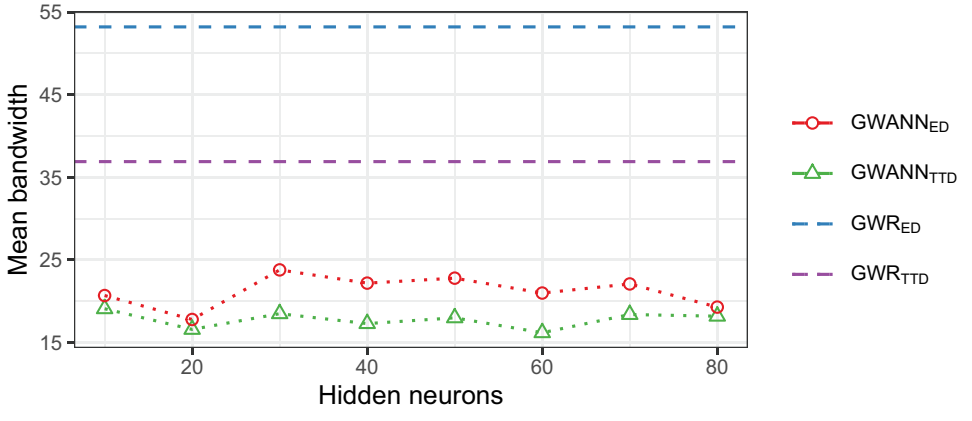

Figure 10. Determined bandwidths (adaptive).

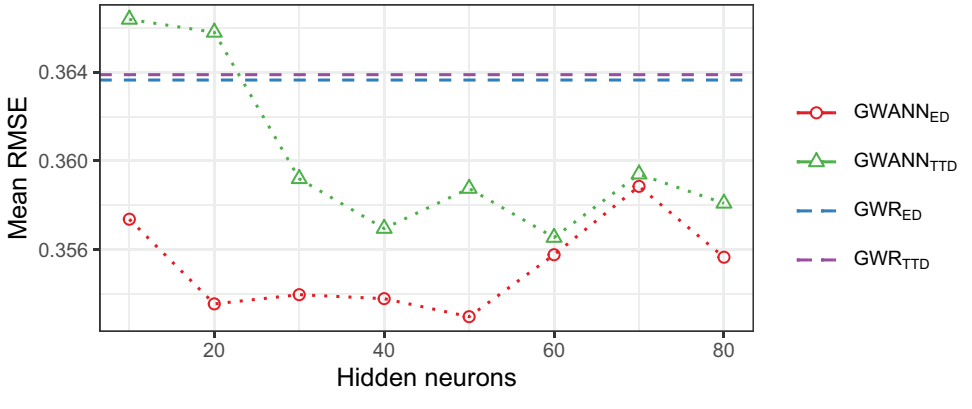

Figure 11. Estimated prediction error (RMSE).

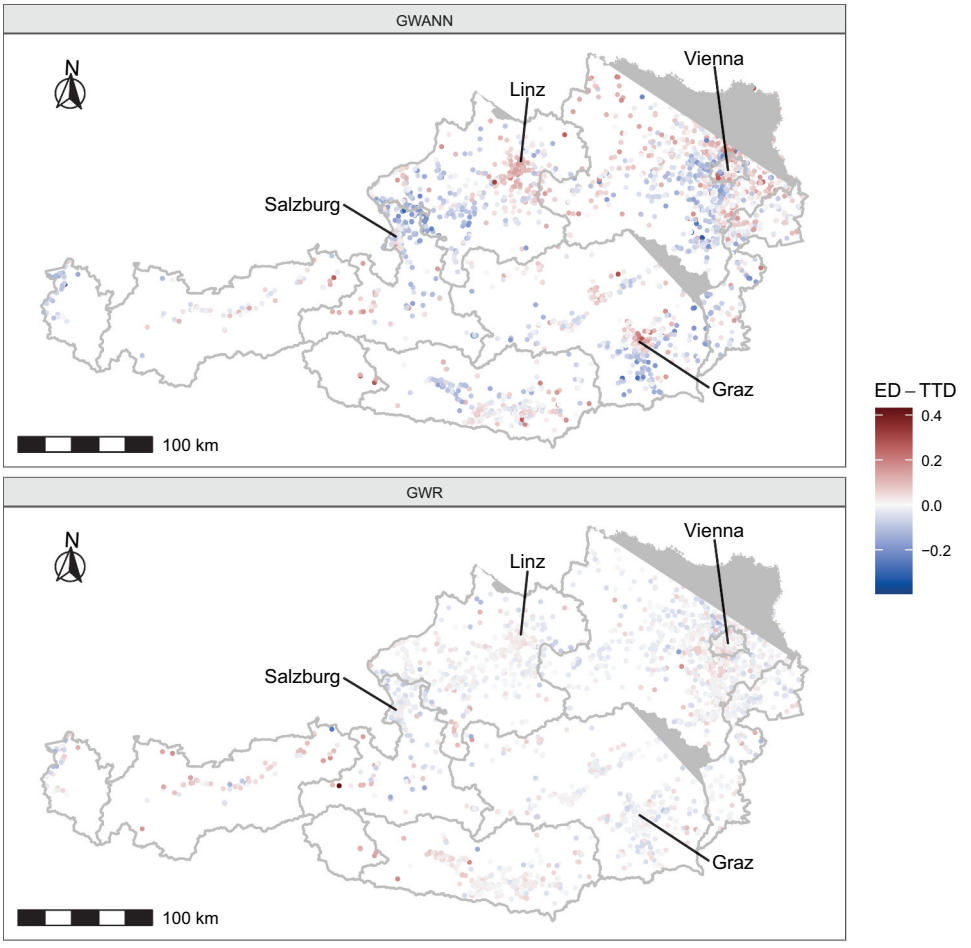

Figure 12. Difference in predicted house prices when using EDs and TTDs for GWR and GWANN. Gray lines demarcate the federal states of Austria.

distance metrics for GWR is barely observable. This confirms the results of Lu *et al.* (2017), who also found no substantial difference in the goodness-of-fit of GWR between EDs and TTDs, and also indicates that the predictive performance of GWANN depends more on the choice of the distance metric than is the case with GWR. Moreover, with the exception of GWANN consisting of fewer than 30 hidden neurons and using TTDs, the mean RMSE of GWANN is always lower than that of GWR, independent of the distance metric used for model building. The overall lowest mean RMSE is obtained by GWANN when using EDs and 60 hidden neurons. The results demonstrate that GWANN can make better predictions than GWR when dealing with spatially heterogeneous relationships in a practical setting.

Using the complete dataset, we built GWR and GWANN using the following hyperparameters. When using EDs, GWR was fitted with a bandwidth of 53 and GWANN was trained with 50 hidden neurons and a bandwidth of 27 for 2,304 iterations. When using

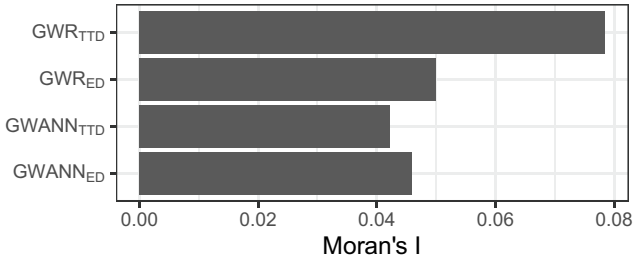

Figure 13. Moran's *I* statistics of the residuals.

TTDs, GWR was fitted with a bandwidth of 36 and GWANN was trained with 60 hidden neurons and a bandwidth of 15 for 2,478 iterations.

To compare the influence of the chosen distance metric on the predictions, Figure 12 shows the differences in predicted house prices when using TTDs and EDs for GWANN and GWR. When using EDs rather than TTDs, GWANN predicts higher house prices for the city of Linz. For the Graz region, a stark contrast between the city and its surroundings is observable: GWANN predicts higher house prices for the city itself but lower house prices for the surroundings when using EDs rather than TTDs. For the metropolitan areas of Vienna, it can be seen that GWANN predicts higher prices in the eastern surroundings of the city and lower prices in the western surroundings when using EDs rather than TTDs. For the city of Salzburg, no differences in predicted house prices are observable. However, in the northern surroundings of Salzburg, substantially lower house prices are predicted when GWANN uses EDs rather than TTDs. For GWR the differences in predicted house prices resulting from the use of either EDs or TTDs are generally small and no spatial patterns are observable. These results demonstrate that in contrast to GWR, for GWANN the choice of the distance metric has a substantial effect on the spatial distribution of the predictions.

Figure 13 shows the Moran's *I* statistics of the models' residuals. For GWR the Moran's *I* values are smaller when using EDs rather than TTDs, while for GWANN they are smaller when using TTDs rather than EDs. Independent of the distance metric, the Moran's *I* values of GWANN are smaller than those of GWR. However, the residuals of both models do not reach statistical significance ($p > 0.05$), suggesting that both models take into account the spatial properties of the data appropriately.

4. Conclusion

We introduced GWANN – a method that combines ANNs and geographical weighting for modeling spatially heterogeneous relationships. We used synthetic and real-world data to compare GWANN with GWR. The results of the synthetic data showed that GWANN can have a better predictive performance than GWR when the relationships within the data are nonlinear and their spatial variance is high. The results based on the real-world data demonstrated that the predictive performance of GWANN can also be superior to that of the competing models in a practical setting.

Notwithstanding these promising results, this study had some limitations that should be considered when interpreting the findings or applying GWANN.

First, the results depended on the choice of the models' hyperparameters. While we followed common practices when choosing the hyperparameters and did careful sensitivity analysis, it cannot be guaranteed that we chose the most appropriate hyperparameters. Comprehensive sensitivity analysis is part of future analysis. Second, while the coefficient surfaces of GWR are useful for analyzing the modeled relationships, the complexity of the computations performed within the network of GWANN makes the interpretation of its surfaces difficult if not impossible. Third, in most practical applications GWANN consists of many output neurons (i.e. one output neuron for each location for which a prediction is to be made). Hence, because each output neuron is connected to each hidden neuron, the number of connection weights can be very large and the adjustment of the connection weights during the training require substantial computational resources. This is particularly a concern when searching for an appropriate bandwidth, which involves training and comparing numerous GWANNs with different bandwidths. More efficient heuristics for finding an appropriate bandwidth have the potential to mitigate this issue. Fourth, in the context of GWR, Fotheringham *et al*. (2017) showed that it is useful to model spatial heterogeneity at different scales by using individual bandwidths for the coefficients. While such an approach also has the potential to improve the predictive performance of GWANN, it remains open to further research as to whether and, if so, how it can be transferred to GWANN.

Note

1. Two additional experiments are given in the supplemental materials. The first one uses housing benchmark data to predict house prices and the second one traffic and land-use data to predict nitrogen dioxide concentrations.

Acknowledgments

We thank the anonymous reviewers for their constructive comments, which greatly improved this article. We also acknowledge UniCredit Bank Austria AG, in particular Wolfgang Brunauer, for providing the housing dataset. The opinions expressed by the authors do not reflect the official viewpoint of UniCredit Bank Austria AG.

Data and codes availability statement

An R package that provides an implementation of GWANN, the source code, and the synthetic datasets can be downloaded from https://github.com/jhagenauer/gwann. The real-estate dataset cannot be shared publicly due to data protection restrictions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research is an outcome of the first author's research stay at Utrecht University in the course of the NEEDS project. The NEEDS project was funded by the European Research Council (ERC) under

the European Union's Horizon 2020 research and innovation program [Grant Agreement No. 714993]. The funders had no role in the study design, data collection and analysis, interpretation, or dissemination.

Notes on contributors

Julian Hagenauer is a data scientist at Daimler AG, Stuttgart. His research intestests are spatial machine learning, spatial data analysis, and how these can be used to solve geospatial problems.

Marco Helbich is associate professor in the Department of Human Geography and Spatial Planning, Utrecht University. His research deals with how the built and natural environments affect human behavior and health outcomes. His research interests focus on geocomputational techniques and spatio-temporal analytics to address human–environment relations in cities.

References

Anselin, L., 1989. *What is special about spatial data? Alternative perspectives on spatial data analysis*. Technical report. Santa Barbara: National Center for Geographic Information and Analysis.

Bárcena, M.J., *et al*., 2014. Alleviating the effect of collinearity in geographically weighted regression. *Journal of Geographical Systems*, 16 (4), 441–466. doi:10.1007/s10109-014-0199-6

Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. *In*: G. Montavon, G. B. Orr and K.-R. Müller, eds. *Neural networks: tricks of the trade*. Berlin: Springer, 437–478.

Bitter, C., Mulligan, G.F., and Dall'erba, S., 2007. Incorporating spatial variation in housing attribute prices: a comparison of geographically weighted regression and the spatial expansion method. *Journal of Geographical Systems*, 9 (1), 7–27. doi:10.1007/s10109-006-0028-7

Brunsdon, C., Fotheringham, A.S., and Charlton, M., 1999. Some notes on parametric significance tests for geographically weighted regression. *Journal of Regional Science*, 39 (3), 497–524. doi:10.1111/0022-4146.00146

Brunsdon, C., Fotheringham, A.S., and Charlton, M.E., 1996. Geographically weighted regression: a method for exploring spatial nonstationarity. *Geographical Analysis*, 28 (4), 281–298. doi:10.1111/j.1538-4632.1996.tb00936.x

Brunsdon, C., Fotheringham, S., and Charlton, M., 2007. Geographically weighted discriminant analysis. *Geographical Analysis*, 39 (4), 376–396. doi:10.1111/j.1538-4632.2007.00709.x

Casetti, E., 1972. Generating models by the expansion method: applications to geographical research. *Geographical Analysis*, 4 (1), 81–91. doi:10.1111/j.1538-4632.1972.tb00458.x

Choi, H. and Kim, H., 2017. Analysis of the relationship between community characteristics and depression using geographically weighted regression. *Epidemiology and Health*, 39, e2017025. doi:10.4178/epih.e2017025

Comber, A., *et al*., 2020. Distance metric choice can both reduce and induce collinearity in geographically weighted regression. *Environment and Planning B: Urban Analytics and City Science*, 47 (3), 489–507.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2 (4), 303–314. doi:10.1007/BF02551274

Du, Z., *et al*., 2020. Geographically neural network weighted regression for the accurate estimation of spatial non-stationarity. *International Journal of Geographical Information Science*, 34 (7), 1353–1377.

Fotheringham, A.S., Brunsdon, C., and Charlton, M., 2002. *Geographically weighted regression: the analysis of spatially varying relationships*. Chichester, UK: Wiley.

Fotheringham, A.S., Crespo, R., and Yao, J., 2015. Geographical and temporal weighted regression (GTWR). *Geographical Analysis*, 47 (4), 431–452. doi:10.1111/gean.12071

Fotheringham, A.S. and Oshan, T.M., 2016. Geographically weighted regression and multi- collinearity: dispelling the myth. *Journal of Geographical Systems*, 18 (4), 303–329. doi:10.1007/s10109-016-0239-5

Fotheringham, A.S., Yang, W., and Kang, W., 2017. Multiscale geographically weighted regression (MGWR). *Annals of the American Association of Geographers*, 107 (6), 1247–1265. doi:10.1080/24694452.2017.1352480

Gorr, W.L. and Olligschlaeger, A.M., 1994. Weighted spatial adaptive filtering: monte Carlo studies and application to illicit drug market modeling. *Geographical Analysis*, 26 (1), 67–87. doi:10.1111/j.1538-4632.1994.tb00311.x

Griffith, D.A., 2003. *Spatial autocorrelation and spatial filtering: gaining understanding through theory and scientific visualization*. Berlin, Heidelberg: Springer Science & Business Media.

Guo, L., Ma, Z., and Zhang, L., 2008. Comparison of bandwidth selection in application of geographically weighted regression: a case study. *Canadian Journal of Forest Research*, 38 (9), 2526–2534. doi:10.1139/X08-091

Hagenauer, J. and Helbich, M., 2018. Local modelling of land consumption in Germany with RegioClust. *International Journal of Applied Earth Observation and Geoinformation*, 65, 46–56. doi:10.1016/j.jag.2017.10.003

Haykin, S., 2008. *Neural networks: a comprehensive foundation*. 3rd ed. Upper Saddle River, NJ: Prentice Hall.

Helbich, M., *et al*., 2014. Spatial heterogeneity in hedonic house price models: the case of Austria. *Urban Studies*, 51 (2), 390–411. doi:10.1177/0042098013492234

Helbich, M. and Griffith, D.A., 2016. Spatially varying coefficient models in real estate: eigen- vector spatial filtering and alternative approaches. *Computers, Environment and Urban Systems*, 57, 1–11. doi:10.1016/j.compenvurbsys.2015.12.002

Huber, S. and Rust, C., 2016. Calculate travel time and distance with OpenStreetMap data using the open source routing machine (OSRM). *The Stata Journal*, 16 (2), 416–423. doi:10.1177/1536867X1601600209

LeSage, J.P. and Pace, R.K., 2009. Introduction to spatial econometrics. *Statistics: A Series of Textbooks and Monographs*. CRC Press.

Leuenberger, M. and Kanevski, M., 2015. Extreme learning machines for spatial environmental data. *Computers & Geosciences*, 85, 64–73. doi:10.1016/j.cageo.2015.06.020

Li, K. and Lam, N.S., 2018. Geographically weighted elastic net: A variable-selection and modeling method under the spatially nonstationary condition. *Annals of the American Association of Geographers*, 108 (6), 1582–1600. doi:10.1080/24694452.2018.1425129

Lu, B., *et al*., 2017. Geographically weighted regression with parameter-specific distance metrics. *International Journal of Geographical Information Science*, 31 (5), 982–998. doi:10.1080/13658816.2016.1263731

Lu, B., Charlton, M., and Fotheringhama, A.S., 2011. Geographically weighted regression using a non-Euclidean distance metric with a study on london house price data. *Procedia Environmental Sciences*, 7, 92–97. doi:10.1016/j.proenv.2011.07.017

Nelson, A., Oberthür, T., and Cook, S., 2007. Multi-scale correlations between topography and vegetation in a hillside catchment of Honduras. *International Journal of Geographical Information Science*, 21 (2), 145–174. doi:10.1080/13658810600852263

Nesterov, Y.E., 1983. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269, 543–547.

Páez, A., Long, F., and Farber, S., 2008. Moving window approaches for hedonic price estimation: an empirical comparison of modelling techniques. *Urban Studies*, 45 (8), 1565–1581. doi:10.1177/0042098008091491

Rojas, R., 2013. *Neural networks: a systematic introduction*. Berlin, Heidelberg: Springer Science & Business Media.

Rosen, S., 1974. Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of Political Economy*, 82 (1), 34–55. doi:10.1086/260169

Rumelhart, D.E., Hinton, G.E., and Williams, R.J., 1986. Learning representations by back- propagating errors. *Nature*, 323 (6088), 533–536. doi:10.1038/323533a0

Sopranzetti, B.J., 2010. Hedonic regression analysis in real estate markets: a primer. *In*: *Handbook of quantitative finance and risk management*. Berlin, Heidelberg: Springer, 1201–1207.

Sunding, D.L. and Swoboda, A.M., 2010. Hedonic analysis with locally weighted regression: an application to the shadow cost of housing regulation in Southern California. *Regional Science and Urban Economics*, 40 (6), 550–573. doi:10.1016/j.regsciurbeco.2010.07.002

Sutskever, I., *et al*. 2013. On the importance of initialization and momentum in deep learning. In: *Proceedings of the 30thInternational conference on machine learning*. Atlanta, Georgia. 1139–1147.

Troy, A., Grove, J.M., and O'Neil-Dunne, J., 2012. The relationship between tree canopy and crime rates across an urban-rural gradient in the greater Baltimore region. *Landscape and Urban Planning*, 106 (3), 262–270. doi:10.1016/j.landurbplan.2012.03.010

Waller, L.A., *et al*., 2007. Quantifying geographic variations in associations between alcohol distribution and violence: a comparison of geographically weighted regression and spatially varying coefficient models. *Stochastic Environmental Research and Risk Assessment*, 21 (5), 573–588. doi:10.1007/s00477-007-0139-9

Wheeler, D.C., 2007. Diagnostic tools and a remedial method for collinearity in geographically weighted regression. *Environment & Planning A*, 39 (10), 2464–2481. doi:10.1068/a38325

Wheeler, D.C., 2009. Simultaneous coefficient penalization and model selection in geographically weighted regression: the geographically weighted lasso. *Environment & Planning A*, 41 (3), 722–742. doi:10.1068/a40256

Wheeler, D.C. and Tiefelsdorf, M., 2005. Multicollinearity and correlation among local regression coefficients in geographically weighted regression. *Journal of Geographical Systems*, 7 (2), 161–187. doi:10.1007/s10109-005-0155-6

Yu, W., *et al*., 2011. Analyzing and modeling land use land cover change (LUCC) in the Daqing City, China. *Applied Geography*, 31 (2), 600–608. doi:10.1016/j.apgeog.2010.11.019