

Chapter 4

Neural Approaches to Economic Modeling

Abstract This chapter describes the multi-layered perceptron (MLP), radial basis functions (RBF), and support vector machines (SVM). These techniques are then applied to model inflation by training these networks using the maximum-likelihood method. The results indicated that the SVM gives the best results followed by the MLP and then the RBF. The SVM with an exponential RBF gave the best results.

4.1 Introduction

This chapter applies the multi-layered perceptron (MLP), radial basis functions (RBF) and support vector machines (SVM) to inflation modeling. It uses economic variables that were identified in Chap. 3 by using the automatic relevance determination method that was based on the MLP. Modeling inflation is an important undertaking for economic planning as it impacts on areas of vital economic activities.

Welfe (2000) applied a Bayesian framework and the Johansen method to model inflation in Poland, while Price and Nasim (1999) applied co-integration and the P-star approach to model inflation and money demand in Pakistan.

De Gooijer and Vidiella-i-Anguera (2003) successfully applied the multiplicative seasonal self-exciting threshold autoregressive which factored monthly inflation rates and seasonal fluctuations at the same time. Gregoriou and Kontonikas (2009) modeled non-linearity in inflation deviations from the target of various countries using an exponential smooth transition autoregressive (ESTAR) and Markov regime-switching models. They observed that the ESTAR model was able to capture the non-linear behavior of inflation. In addition they observed a fast adjustment process in countries that regularly under estimate inflation target while countries that over-estimate inflation target displayed a slower revision back to equilibrium. The ESTAR model was also observed to outperform the Markov regime-switching model.

Aïssa et al. (2007) modelled inflation persistence with periodicity changes in fixed and pre-determined prices models. In particular, they considered the relationship between the length of the decision interval for a specified duration of prices and the dynamic properties of inflations. They observed that higher frequency types of information, overlapping contracts and hybrid price models forecast inflation persistence.

Mehrotra et al. (2010) applied the hybrid New Keynesian Phillips Curve (NKPC) for modelling inflation in China and demonstrated that the NKPC gave good results of inflation. A probit analysis showed that the forward-looking inflation element and the output gap were vital inflation variables in provinces that had undergone market reform.

Kukul and Van Quang (2011) applied the multi-layered perceptron and radial basis function neural networks to model how Czech National Bank handled its repo rate when it implemented its monetary policy and observed that the MLP network was better at modeling the repo rate than the RBF.

Choudhary and Haider (2012) compared neural networks to autoregressive models for inflation forecasting and found that neural networks performed better than the autoregressive models.

Düzgün (2010) compared the generalized regression neural networks to the autoregressive integrated moving average (ARIMA) and found that neural network model performed better than the ARIMA model in forecasting the Consumer Price Index.

Santana (2006) compared neural networks to exponential smoothing and ARIMA methods for forecasting Colombian Inflation and observed that neural network forecasts were more accurate than the forecasts from exponential smoothing and ARIMA methods.

Binner et al. (2005) compared linear forecasting models and neural networks for modeling Euro inflation rate. The results obtained indicated that neural networks gave better forecasts than linear models.

Moshiri and Cameron (2000) compared neural network to econometric models for forecasting inflation and they observed that neural networks were able to forecast marginally better than traditional econometric methods.

Other approaches that have been used to model inflation are support vector machines (Wang et al. 2012), genetic optimization (Huang 2012a), neural networks (Wang and Fan 2010; Ge and Yin 2012), dynamic factor estimation method (Kapetanios 2004), combinatorial and optimal networks (Luo and Huang 2012), and elitist neuro-genetic network model (Mitra 2012). Other aspects of inflation that have been modeled are the relationship between short-term and long term inflation expectations (Jochmann et al. 2010), pass-through of individual goods prices into trend inflation (Anderson et al. 2012), identification of post-stabilization inflation dynamics (Dimirovski and Andreeski 2006) and medical costs inflation rates (Cao et al. 2012).

From this literature review, it is evident that artificial intelligence methods are viable tools for modelling inflation. The next sections will describe multi-layered

perceptrons, radial basis functions, and support vector machines which are techniques that are used to model inflation in this chapter.

4.2 Multi-layer Perceptron Neural Networks

The neural network model considered in this chapter is the multi-layer perceptron. The MLP is a feed-forward neural network model that estimates a relationship between sets of input data and a set of corresponding output. Its basis is the standard linear perceptron and it uses three or more layers of neurons, also called nodes, with non-linear activation functions. It is more effective than the perceptron because it can discriminate data that is not linearly separable or separable by a hyper-plane.

The multi-layered perceptron has been used to model many diverse complex systems in areas such as mechanical engineering (Marwala 2010, 2012), missing data (Marwala 2009), interstate conflict (Marwala and Lagazio 2011), soil water assessment (Singh et al. 2012), solid waste (Arebey et al. 2012), in biomarkers to infer characteristics of contaminant exposure (Karami et al. 2012), collision avoidance of a ship (Ahn et al. 2012), speech recognition (Ikbali et al. 2012), and power transformers fault diagnosis (Souahlia et al. 2012). More relevant to the topic of economic modeling, Tung et al. (2004) applied neural networks to build an early warning system for banking failure, while Yümlü et al. (2005) used neural networks for Istanbul stock exchange prediction. Tsai and Wu (2008) applied ensembles of multi-layer perceptrons for bankruptcy prediction and credit scoring, and observed that the ensemble performed better than the stand-alone networks.

As described in Chap. 3, the MLP neural network is made up of multiple layers of computational components typically inter-connected in a feed-forward way (Haykin 1999; Hassoun 1995; Marwala 2012). Every neuron in one layer is linked to the neurons of the following layer. A completely linked two layered MLP network is used in this chapter. A NETLAB[®] toolbox that runs in MATLAB[®] which was built by Nabney (2001) was implemented to build the MLP neural network. The network which was discussed in Chap. 3 can be mathematically described as follows (Haykin 1999):

$$y_k = f_{outer} \left(\sum_{j=1}^M w_{kj}^{(2)} f_{inner} \left(\sum_{i=1}^d w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)} \right) \quad (4.1)$$

Here $w_{ji}^{(1)}$ and $w_{ji}^{(2)}$ are weights in the first and second layer, respectively, from input i to hidden unit j , M is the number of hidden units, d is the number of output units while $w_{j0}^{(1)}$ and $w_{k0}^{(2)}$ are the weight parameters that represent the biases for the hidden unit j and the output unit k . These weight parameters can be interpreted as an instrument that ensures that the model comprehends the data. In this chapter, the parameter $f_{outer}(\bullet)$ is the linear function while f_{inner} is the hyperbolic tangent

function. As described in Chap. 3, the weight vector in Eq. 4.1 is identified using the scaled conjugate gradient method that is premised on the maximum-likelihood approach (Møller 1993).

4.3 Radial-Basis Function (RBF)

The other neural network technique that is applied in this chapter is the radial basis function. The RBF neural networks are feed-forward networks which are trained using a supervised training algorithm (Haykin 1999; Buhmann and Ablowitz 2003; Marwala and Lagazio 2011).

Marcozzi et al. (2001) successfully applied RBF and finite-difference algorithms for option pricing, while Jing and Ning-qiang (2012) applied the RBF in power system's damping analysis. Guo et al. (2012) successfully applied RBF to predict financial risk of the thermal power industry. Other successful applications of the RBF to model complex problems include controlling chaotic systems (Hsu et al. 2012), braking system of a railway vehicle (Yang and Zhang 2012) and prediction of protein interaction sites (Chen et al. 2012b).

Golbabai and Milev (2012) solved a partial integro-differential equation problem with a free boundary in an American option model. They applied front-fixing transformation of the underlying asset variable to deal with the free boundary conditions. A radial basis functions technique was applied to realize a first order nonlinear equation and they used the Predictor–Corrector technique to solve the nonlinear equations. The results obtained were similar to those achieved by other numerical methods.

Yu et al. (2012) applied a hybrid of particle swarm optimization and RBF (PSO-RBF) for predicting China's primary energy demands in 2020. They analyzed the energy demand for the period from 1980 to 2009 based on the gross domestic product (GDP), population, proportion of industry in GDP, urbanization rate, and share of coal energy. The results they obtained showed that the PSO–RBF had fewer hidden nodes and smaller estimated errors than traditional neural network models.

The RBF is normally arranged with a single hidden layer of units whose activation function is chosen from a class of functions known as the basis functions. The activation of the hidden units in an RBF neural network is described by a non-linear function of the distance between the input vector and a vector representing the centers of mass of the data (Bishop 1995). Although related to back-propagation radial basis function networks have a number of advantages. They train much faster than the MLP networks and are less disposed to problems with non-stationary inputs owing to the behavior of the radial basis function (Bishop 1995). The RBF network is shown in Fig. 4.1 and is expressed mathematically as follows (Buhmann and Ablowitz 2003; Marwala and Lagazio 2011):

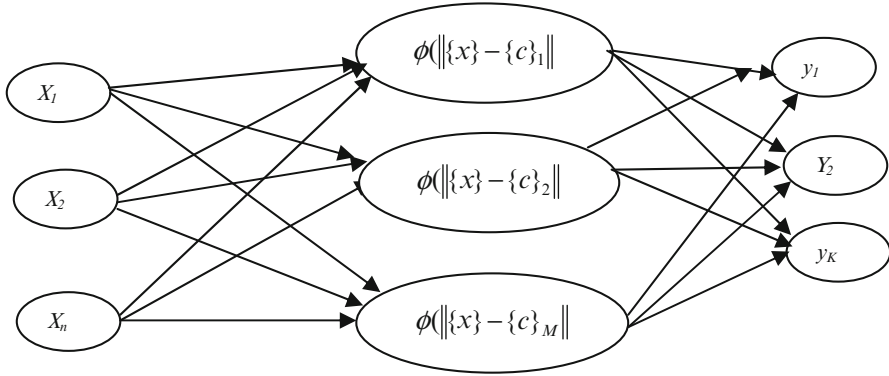


Fig. 4.1 Radial basis function network having two layers of Adaptive weights (Reprinted from Marwala and Lagazio 2011 with kind permission from Springer Science + Business Media B.V)

$$y_k(\{x\}) = \sum_{j=1}^M w_{jk} \phi(\|\{x\} - \{c\}_j\|) \quad (4.2)$$

Here, w_{jk} are the output weights, corresponding to the link between a hidden unit and an output unit, M denotes the number of hidden units, $\{c\}_j$ is the center for the j th neuron, $\phi(\{x\})$ is the j th non-linear activation function, $\{x\}$ the input vector, and $k = 1, 2, 3, \dots, M$ (Bishop 1995; Marwala and Lagazio 2011). Once more, as in the MLP, the choice of the number of hidden nodes M is part of the model selection procedure.

The activation in the hidden layers implemented in this chapter is a Gaussian distribution which can be written as follows (Bishop 1995):

$$\left(\phi(\|\{x\} - \{c\}\|) = \exp\left(-\beta(\{x\} - \{c\})^2\right) \right) \quad (4.3)$$

where β is constant.

As explained by Marwala (2009), the radial basis function is different from the multi-layered perceptron because it has weights in the outer layer only and the hidden nodes have centers. Training the radial basis function network involves identifying two sets of parameters and these are the centers and the output weights and both of these can be regarded as free parameters in a regression framework. Although the centers and network weights can both be estimated at the same time, in this chapter we apply a two stage training process to identify the centers. The first is to use self-organizing maps to estimate the centers and, in this chapter, we apply the k -means clustering technique (Hartigan 1975; Marwala 2009). The step of identifying the centers only considers the input space whereas the identification of the network weights uses both the input and output space.

The *k-means* procedure is intended for clustering objects based on features into k partitions. In this chapter, k will be equal to the number of centers M . Its aim is to determine the centers of natural clusters in the data and assumes that the object features form a vector space. It realizes this by minimizing the total intra-cluster variance, or, the squared error function (Hartigan and Wong 1979; Marwala 2009):

$$E = \sum_{i=1}^C \sum_{x_j \in S_i} (\{x\}_j - \{c\}_i)^2 \quad (4.4)$$

Here, C is the number of clusters S_i , $i = 1, 2, \dots, M$ and $\{c\}_i$ is the centers of all the points $x_j \in S_i$. The Lloyd procedure is applied to determine the cluster centers (Lloyd 1982).

As described by Lloyd (1982), the procedure is started by randomly segmenting the input space into k sets. Then the mean point is computed for each set, and then a new partition is created by linking each point with the nearest center. The centroids are then recomputed for the new clusters, and the procedure is reiterated by varying these two steps until convergence. Convergence is realized when the centroids no longer change or the points no longer switch clusters.

After the centers have been determined, the network weights then need to be estimated, given the training data. To realize this, the Moore-Penrose pseudo inverse (Moore 1920; Penrose 1955; Golub and Van Loan 1996; Marwala 2009) is applied. Once the centers have been determined, then the approximation of the network weights is a linear process (Golub and Van Loan 1996). With the training data and the centers determined, Eq. 4.2 can then be rewritten as follows (Marwala and Lagazio 2011):

$$[y_{ij}] = [\phi_{ik}] [w_{kj}] \quad (4.5)$$

where $[y_{ij}]$ is the output matrix with i denoting the number of training examples, and j denoting the number of output instances. The parameter $[\phi_{ik}]$ is the activation function matrix in the hidden layer, with i denoting the training examples, and k denoting the number of hidden neurons. The parameter $[w_{kj}]$ is the weight matrix, with k denoting the number of hidden neurons and j denoting the number of output instances. It can thus be observed that to estimate the weight matrix $[w_{kj}]$, the activation function matrix $[\phi_{ik}]$ ought to be inverted. Nevertheless, this matrix is not square and consequently it can be inverted using the Moore-Penrose pseudo-inverse is as follows (Marwala and Lagazio 2011):

$$[\phi_{ik}]^* = \left([\phi_{ik}] [\phi_{ik}]^T \right)^{-1} [\phi_{ik}]^T \quad (4.6)$$

This means that the weight matrix may be estimated as follows (Marwala and Lagazio 2011):

$$[w_{kj}] = [\phi_{ik}]^* [y_{ij}] \quad (4.7)$$

A NETLAB[®] toolbox that runs in MATLAB[®] and explained in (Nabney 2001) was applied to implement both the RBF and the MLP architectures.

4.3.1 Model Selection

Given that techniques of determining the weights that are learned from the training data given an RBF or even an MLP model, the following undertaking is how to select a suitable model, given by the size of the hidden nodes. The procedure of choosing suitable model is called as model selection (Burnham and Anderson 2002). The idea of deriving a model from data is a no-unique problem. This is for the reason that many models are capable of describing the training data and, consequently, it becomes difficult to determine the most suitable model. The basic method for model selection is based on two principles and these are the goodness of fit and complexity. The goodness of fit, basically, suggests that a good model should be capable of predicting the validation data which has not been used to train the networks. The complexity of the model is actually based on the principle of Occam's razor that recommends that the preferred model should be the simplest one.

There are vital issues such as:

- How do you choose a compromise between the goodness of fit and the complexity of the model?
- How are these characteristics applied?

The goodness of fit is evaluated, in this chapter, by calculating the error between the model prediction and the validation set whereas the complexity of the model is evaluated by the number of free parameters in the data. As explained by Marwala (2009), free parameters in the MLP model are defined as the network weights and biases whereas in the RBF network are defined as the network centers and weights. In this chapter, model selection is regarded as an instrument of choosing a model that has a good probability of approximating the validation data that it has not been exposed to during network training and the bias and variance are indicators of the capability of this model to function in an acceptable manner.

The MLP is trained in this chapter using the scaled conjugate gradient technique while for the RBF the *k-means* and pseudo-inverse techniques are used estimate the free parameters (centers and weights).

4.4 Support Vector Regression

The type of artificial intelligence method called support vector machines are a supervised learning technique applied principally for classification and are resulting from statistical learning theory and were first presented by Vapnik (1995, 1998). They have also been modified to handle regression problems (Gunn 1997; Chang and Tsai 2008; Chuang 2008; Hu et al. 2012; Orchel 2012). As described in Marwala (2009), Pires and Marwala (2004) successfully applied support vector machines for option pricing and additionally extended these to the Bayesian framework whereas Gidudu et al. (2007) successfully applied support vector machines for image classification.

Furthermore, Khemchandani et al. (2009) successfully applied the regularized least squares fuzzy support vector regression for financial time series forecasting. Chen et al. (2012a) successfully used support vector machines for seismic assessment of school buildings, while Golmohammadi et al. (2012) used support vector machines for quantitative structure–activity relationship estimation of blood-to-brain partitioning behavior. Zhang et al. (2006) successfully used support vector regression for on-line health monitoring of large-scale structures whereas Chen et al. (2012c) applied SVM for customer churn prediction.

One of the difficulties of support vector regression is the computational load required to train them. Guo and Zhang (2007) established methods for accelerating support vector regression whereas Üstün et al. (2007) visualized support vector regression models. Other successful applications of support vector machine include prediction of carbon monoxide concentrations (Yeganeh et al. 2012), bearing fault detection in industrial environments (Gryllias and Antoniadis 2012), defect recognition in SonicIR (Zeng et al. 2012), classification of small-grain weed species (Rumpf et al. 2012), crash injury severity analysis (Li et al. 2012) and mechanical fault condition monitoring in induction motor (Salem et al. 2012). Huang (2012b) applied genetic algorithms and support vector regression for hybrid stock selection model and demonstrated that the proposed method gave better investment returns than the benchmark.

The general notion behind support vector regression is to relate the input space to an output space. Assuming we have the training dataset with one input and one output being taken into account: $\{(x_1, y_1), \dots, (x_i, y_i)\} \subset \chi \times \Re$, where χ is the space of the input parameters and \Re represents the real number set. We intend to identify a function $f(x)$ that will relate the training inputs to the training outputs. In Support Vector regression the objective is to identify this function that has at most ε deviation from the actual training targets y_i . We can identify a number of functions $f(x)$ to relate training inputs to training outputs. These functions are called kernel functions nevertheless these cannot just be any functions because kernel functions have to satisfy to some principles (Joachims 1999). As described by many other researchers we study a linear kernel function (Lang 1987):

$$f(x) = \langle w, x \rangle + b \quad \text{with} \quad w \in \chi, \quad b \in \Re \quad (4.8)$$

Table 4.1 Common types of kernels

Kernel Type	Equation
Linear	$k(x_i, x_j) = (x_i \cdot x_j)$
Polynomial (inhomogeneous)	$k(x_i, x_j) = (x_i \cdot x_j + c)^d$ where $d \geq 2$ and $c > 0$
Radial basis function	$k(x_i, x_j) = \exp(-\gamma \ x_i - x_j\ ^2)$ for $\gamma > 0$
Exponential radial basis function	$k(x_i, x_j) = \exp\left(-\frac{1}{2\sigma^2} \ x_i - x_j\ ^2\right)$
Spline	$k(x_i, x_j) = 1 + x_i x_j + \frac{1}{2} x_i - x_j \min(x_i, x_j)^2 + \frac{\min(x_i, x_j)^3}{3}$

where (\cdot, \cdot) represents the dot product. Other kernel functions that can be used include exponential radial basis function, linear spline, radial basis function and polynomial which are shown in Table 4.1.

It is intended to identify small values for w and this can be achieved by minimizing the Euclidean norm $\|w\|^2$ (Drezet and Harrison 2001). A slack variables ξ_i, ξ_i^* can be included in order to ensure that particular infeasible constraints in the minimization of the Euclidean norm can be applied and the minimization problem then becomes (Oliveira 2006):

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \tag{4.9}$$

$$\text{subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{4.10}$$

where l is the number of training points used. The constraints in Eq. 4.10 deal with an ε -insensitive loss function applied to penalize particular training points that are outside of the bound given by ε which is a value selected.

Some other loss functions that may be used include smooth non-convex loss function (Zhong 2012) and the Huber loss function which can also be applied but the most common one is the ε -insensitive loss function which is given by (Gunn 1997):

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases} \tag{4.11}$$

As explained in Marwala (2009), the value for C is the extent to which deviations from ε are accepted (Trafalis and Ince 2000). It can be viewed as a degree of overfitting a function to the training data. If the value of C is set too high then the function $f(x)$ will be too well fitted to the training data and will not forecast well on data that

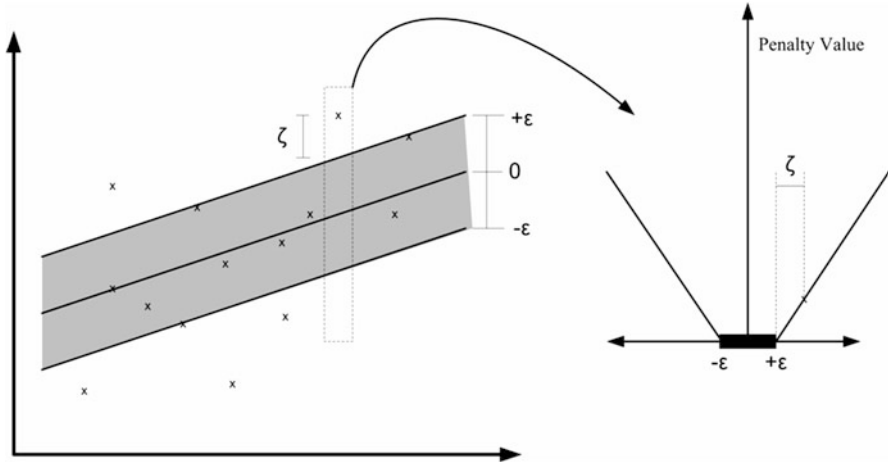


Fig. 4.2 Linear support vector regression for a set of data (left) and the ϵ -insensitive loss function (right) (Adapted from Gunn 1997)

has not seen by the training of the function. This implies that data located outside of the bounds given by ϵ are not penalized sufficiently resulting in the function being too well fitted to the training data (Trafalis and Ince 2000). An illustration of a linear function being fitted to the training data can be observed in Fig. 4.2 with the bounds displayed.

The function in Fig. 4.2 (right hand side) is applied to penalize those data points that are located outside of the bounds (left hand side). The further a data point is located outside of one of the bounds, the more the data point is penalized and, therefore, it is factored less in the estimation of the function. Those data points that are located within the bounds of the function are not penalized at all and their matching slack variable values (ξ_i, ξ_i^*) are assigned zero and accordingly these data points contribute significantly in the estimation of the function $f(x)$.

The optimization problem in Eq. 4.9 is then formulated as a quadratic programming problem by first determining the Lagrangian multiplier and using the Karush-Kuhn Tucker (KKT) conditions (Joachims 1999). Then the values for w and b can be estimated so that the linear function fit the training data can be identified. This instance of applying the constrained optimization framework is merely applicable for a linear kernel function and the constrained optimization framework differs for different kernel functions.

Similarly, a non-linear model can be used to satisfactorily model the data and this can be conducted by applying a non-linear function to relate the data into a high dimensional feature space where linear regression is conducted. Then the kernel method is applied to handle the problem of the curse of dimensionality and for non-linear problems the ϵ -insensitive loss function can be used to give (Gunn 1997):

$$\begin{aligned} \max_{\alpha, \alpha^*} W(\alpha, \alpha^*) = & \max_{\alpha, \alpha^*} \sum_{i=1}^l \alpha^* (y_i - \varepsilon) \\ & - \alpha_i (y_i + \varepsilon) - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) K(x_i, x_j) \end{aligned} \quad (4.12)$$

subject to

$$\begin{aligned} 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, \dots, l \\ \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \end{aligned} \quad (4.13)$$

In Eqs. 4.12 and 4.13, K is the kernel function and α and α^* are Lagrangian multipliers. Solving these equations provides the Lagrangian multipliers and the regression equation can thus be written as follows (Gunn 1997):

$$f(x) = \sum_{SVs} (\bar{\alpha}_i - \bar{\alpha}_i^*) K(x_i, x) + \bar{b} \quad (4.14)$$

$$b = -\frac{1}{2} \sum_{i=1}^l (\alpha_i - \alpha_i^*) (K(x_i, x_r) + K(x_i, x_s)) \quad (4.15)$$

The least squares support vector toolbox was applied for the investigation (Suykens et al. 2002).

4.5 Applications of MLP, RBF and SVM to Economic Modeling

In Chap. 3, the Bayesian and the evidence frameworks were applied to build an automatic relevance determination (ARD) method. The ARD technique was then applied to identify the relevance of economic variables that are essential for driving the consumer price index. It was concluded that for the data analyzed using the MLP based ARD the variables driving the CPI are mining, transport, storage and communication, financial intermediation, insurance, real estate and business services, community, social and personal services, gross value added at basic prices, taxes less subsidies on products, affordability, economic growth, repo rate, gross domestic product, household consumption and investment.

Table 4.2 Characteristics of the MLP network and the results

Attributes	Number
Input nodes	12
Hidden nodes	8
Output nodes	1
Training time	3.45
Accuracy	84.6 %

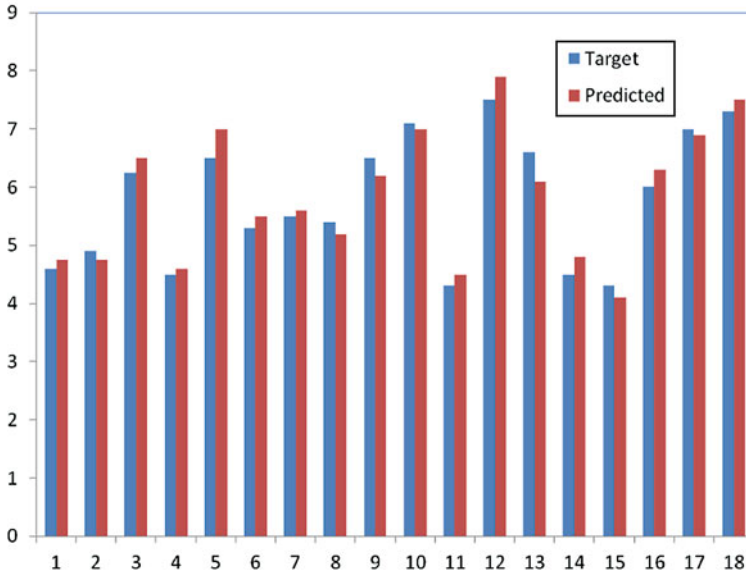


Fig. 4.3 Sample results achieved using the MLP network

Table 4.3 Characteristics of the RBF network and the results

Attributes	Number
Input nodes	12
Hidden nodes	8
Output nodes	1
Training time	0.36
Accuracy	82.7 %

In this chapter, these variables are used to predict the CPI using MLP, RBF, and SVM. The architecture of the MLP neural network applied in this chapter is as indicated in Table 4.2. The MLP had linear activation function in the output layer and hyperbolic tangent function in the hidden nodes (Fig. 4.3). The architecture of the RBF neural network, applied in this chapter, is as indicated in Table 4.3 and Fig. 4.4. The RBF had a Gaussian basis function. The sample results obtained can be viewed in Fig. 4.3.

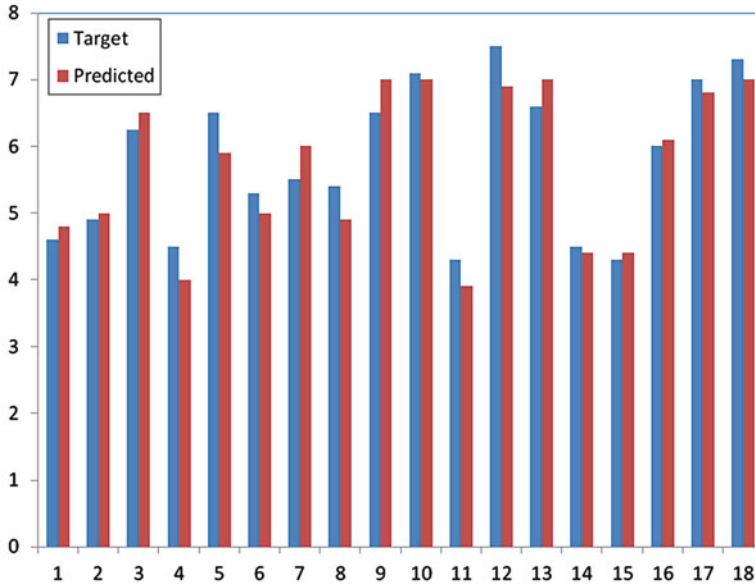


Fig. 4.4 Sample results achieved using the RBF network

Table 4.4 Different implementations of the SVM models

	C	e	Order	Kernel	Accuracy
Model 1	200	0.05	NA	Exponential RBF	89.39
Model 2	200	0.05	NA	Linear	87.81
Model 3	200	0.05	1	Spline	83.45
Model 4	200	0.05	2	RBF	85.73
Model 5	200	0.05	3	Polynomial	88.36

The architectures of the different SVM networks applied in this chapter are as indicated in Table 4.4. The C and e parameters were fixed at 200 and 0.05 respectively.

Figure 4.5 shows the sample results from the RBF based SVM.

The results indicate that for the data under consideration, the SVM models demonstrated better results followed by the MLP then the RBF. The SVM with the exponential RBF gave the best results.

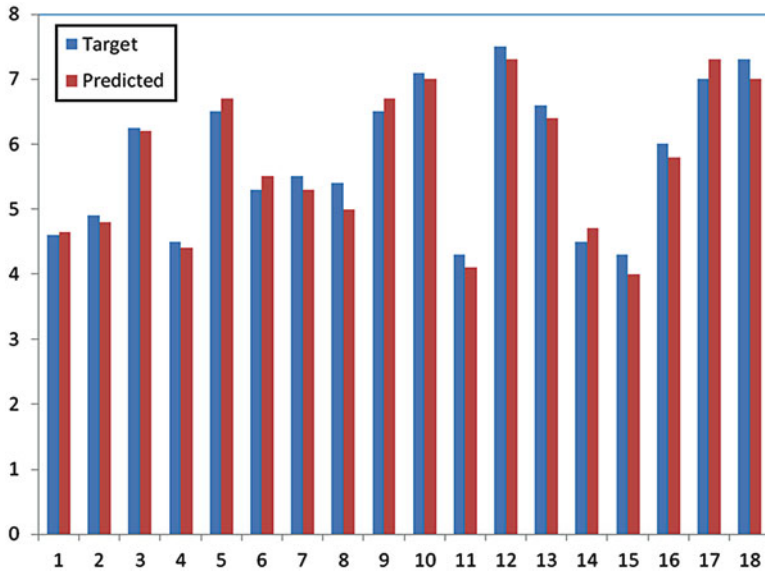


Fig. 4.5 Model 1 predicted CPI vs. the actual CPI

4.6 Conclusion

This chapter described the multi-layered perceptron, radial basis functions, and support vector machines and applied these to modeling the CPI. The results indicated that the SVM gave the best results followed by the MLP and then the RBF. The SVM with an exponential RBF gave the best results.

References

- Ahn J-H, Rhee K-P, You Y-J (2012) A study on the collision avoidance of a ship using neural networks and fuzzy logic. *Appl Ocean Res* 37:162–173
- Aïssa MBS, Musy O, Pereau J-C (2007) Modelling inflation persistence with periodicity changes in fixed and predetermined prices models. *Econ Model* 24:823–838
- Anderson RG, Binner JM, Schmidt VA (2012) Connectionist-based rules describing the pass-through of individual goods prices into trend inflation in the United States. *Econ Lett* 117:174–177
- Arebey M, Hannan MA, Begum RA, Basri H (2012) Solid waste bin level detection using gray level co-occurrence matrix feature extraction approach. *J Environ Manag* 104:9–18
- Binner JM, Bissoondeal RK, Elger T, Gazely AM, Mullineux AW (2005) A comparison of linear forecasting models and neural networks: an application to Euro inflation and Euro Divisia. *Appl Econ* 37:665–680
- Bishop CM (1995) *Neural networks for pattern recognition*. Oxford University Press, Oxford
- Buhmann MD, Ablowitz MJ (2003) *Radial basis functions: theory and implementations*. Cambridge University, Cambridge

- Burnham KP, Anderson DR (2002) Model selection and multimodel inference: a practical information-theoretic approach, 2nd edn. Springer, New York
- Cao Q, Ewing BT, Thompson MA (2012) Forecasting medical cost inflation rates: a model comparison approach. *Decis Support Syst* 53:154–160
- Chang BR, Tsai HF (2008) Forecast approach using neural network adaptation to support vector regression grey model and generalized auto-regressive conditional heteroskedasticity. *Expert Syst Appl* 34:925–934
- Chen C-S, Cheng M-Y, Wu Y-W (2012a) Seismic assessment of school buildings in Taiwan using the evolutionary support vector machine inference system. *Expert Syst Appl* 39:4102–4110
- Chen Y, Xu J, Yang B, Zhao Y, He W (2012b) A novel method for prediction of protein interaction sites based on integrated RBF neural networks. *Comput Biol Med* 42:402–407
- Chen Z-Y, Fan Z-P, Sun M (2012c) A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *Eur J Oper Res* 223:461–472
- Choudhary M, Haider A (2012) Neural network models for inflation forecasting: an appraisal. *Appl Econ* 44:2631–2635
- Chuang C-C (2008) Extended support vector interval regression networks for interval input–output data. *Inf Sci* 178:871–891
- De Gooijer JG, Vidiella-i-Anguera A (2003) Nonlinear stochastic inflation modelling using SEASETARs. *Insur Math Econ* 32:3–18
- Dimirovski GM, Andreeski CJ (2006) How good ANN identification of post-stabilization inflation dynamics can be? In: Proceedings of the IEEE international conference on neural networks, Vancouver, pp 2098–2105
- Drezet PML, Harrison RF (2001) A new method for sparsity control in support vector classification and regression. *Pattern Recognit* 34:111–125
- Düzgün R (2010) Generalized regression neural networks for inflation forecasting. *Int Res J Finance Econ* 51:59–70
- Ge L, Yin G (2012) Application of process neural network on consumer price index prediction. *Adv Intell Soft Comput* 137:427–432
- Gidudu A, Hulley G, Marwala T (2007) Image classification using SVMs: one-against-one vs one-against-all. In: Proceedings of the 28th Asian conference on remote sensing, Kuala Lumpur, arXiv:0711.2914v1
- Golbabai DA, Milev M (2012) Radial basis functions with application to finance: american put option under jump diffusion. *Math Comput Model* 55:1354–1362
- Golmohammadi H, Dashtbozorgi Z, Acree WE Jr (2012) Quantitative structure–activity relationship prediction of blood-to-brain partitioning behavior using support vector machine. *Eur J Pharm Sci* 47:421–429
- Golub GH, Van Loan CF (1996) Matrix computations. Johns Hopkins University Press, Baltimore
- Gregoriou A, Kontonikas A (2009) Modeling the behaviour of inflation deviations from the target. *Econ Model* 26:90–95
- Gryllias KC, Antoniadis IA (2012) A Support Vector Machine approach based on physical model training for rolling element bearing fault detection in industrial environments. *Eng Appl Artif Intell* 25:326–344
- Gunn SR (1997) Support vector machines for classification and regression. Technical report, University of Southampton, Southampton
- Guo G, Zhang JS (2007) Reducing examples to accelerate support vector regression. *Pattern Recognit Lett* 28:2173–2183
- Guo X, Wang H, Yang F (2012) Thermal power financial environment risk forecast model by combined stock multi-indicators basis on RBF neural network. *AASRI Procedia* 1:519–524
- Hartigan JA (1975) Clustering algorithms. Wiley, New York
- Hartigan JA, Wong MA (1979) A K-means clustering algorithm. *Appl Stat* 28:100–108
- Hassoun MH (1995) Fundamentals of artificial neural networks. MIT Press, Cambridge, MA
- Haykin S (1999) Neural networks. Prentice-Hall, Upper Saddle River
- Hsu C-F, Chiu C-J, Tsai J-Z (2012) Indirect adaptive self-organizing RBF neural controller design with a dynamical training approach. *Expert Syst Appl* 39:564–573

- Hu T, Xiang D-H, Zhou D-X (2012) Online learning for quantile regression and support vector regression. *J Stat Plan Inference* 142:3107–3122
- Huang C-F (2012a) A hybrid stock selection model using genetic algorithms and support vector regression. *Appl Soft Comput J* 12:807–818
- Huang S (2012b) Based on genetic optimization of support vector machine consumer price index forecast. *Adv Sci Lett* 6:485–488
- Ikbal S, Misra H, Hermansky H, Magimai-Doss M (2012) Phase AutoCorrelation (PAC) features for noise robust speech recognition. *Speech Commun* 54:867–880
- Jing B, Ning-qiang J (2012) Power system's damping analysis based on RBF neural network1. *Phys Procedia* 24:1018–1023
- Joachims J (1999) Making large-scale SVM learning practical. In: Scholkopf B, Burges CJC, Smola AJ (eds) *Advances in kernel methods-support vector learning*. MIT Press, Cambridge
- Jochmann M, Koop G, Potter SM (2010) Modeling the dynamics of inflation compensation. *J Empir Finance* 17:157–167
- Kapetanios G (2004) A note on modelling core inflation for the UK using a new dynamic factor estimation method and a large disaggregated price index dataset. *Econ Lett* 85:63–69
- Karami A, Christianus A, Bahraminejad B, Gagné F, Courtenay SC (2012) Artificial neural network modeling of biomarkers to infer characteristics of contaminant exposure in *Clarias gariepinus*. *Ecotoxicol Environ Saf* 77:28–34
- Khemchandani R, Jayadeva K, Chandra S (2009) Regularized least squares fuzzy support vector regression for financial time series forecasting. *Expert Syst Appl* 36:132–138
- Kukal J, Van Quang T (2011) Modeling the CNB's monetary policy interest rate by artificial neural networks [Modelování Měnově Politické Úrokové Míry ČNB Neuronovými Síťemi]. *Politická Ekonomie* 59:810–829
- Lang S (1987) *Linear algebra*. Springer, London
- Li Z, Liu P, Wang W, Xu C (2012) Using support vector machine models for crash injury severity analysis. *Accid Anal Prev* 45:478–486
- Lloyd SO (1982) Least squares quantization in PCM. *IEEE Trans Inf Theory* 28:129–137
- Luo F, Huang S (2012) The application of the combinatorial and optimal networks on the prediction of the consumer price index. *Adv Mater Res* 488–489:886–891
- Marcozzi MD, Choi S, Chen CS (2001) On the use of boundary conditions for variational formulations arising in financial mathematics. *Appl Math Comput* 124:197–214
- Marwala T (2009) *Computational intelligence for missing data imputation, estimation and management: knowledge optimization techniques*. IGI Global Publications, New York
- Marwala T (2010) *Finite element model updating using computational intelligence techniques*. Springer, London
- Marwala T (2012) *Condition monitoring using computational intelligence methods*. Springer, London
- Marwala T, Lagazio M (2011) *Militarized conflict modeling using computational intelligence techniques*. Springer, London
- Mehrotra A, Peltonen T, Santos Rivera A (2010) Modelling inflation in china – a regional perspective. *China Econ Rev* 21:237–255
- Mitra S (2012) Early warning prediction system for high inflation: an elitist neuro-genetic network model for the Indian economy. *Neural Comput Appl*. doi:10.1007/s00521-012-0895-4
- Møller AF (1993) A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw* 6:525–533
- Moore EH (1920) On the reciprocal of the general algebraic matrix. *Bull Am Math Soc* 26:394–395
- Moshiri S, Cameron N (2000) Neural network versus econometric models in forecasting inflation. *J Forecast* 19:201–217
- Nabney IT (2001) *Netlab algorithms for pattern recognition*. Springer, London
- Oliveira ALI (2006) Estimation of software project effort with support vector regression. *Neuro-computing* 69:1749–1753
- Orchel M (2012) Support vector regression based on data shifting. *Neurocomputing* 96:2–11
- Penrose R (1955) A generalized inverse for matrices. *Proc Camb Philos Soc* 51:406–413

- Pires MM, Marwala T (2004) Option pricing using neural networks and support vector machines. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, The Hague, 2004, pp 1279–1285
- Price S, Nasim A (1999) Modelling inflation and the demand for money in Pakistan; cointegration and the causal structure. *Econ Model* 16:87–103
- Rumpf T, Römer C, Weis M, Sökefeld M, Gerhards R, Plümer L (2012) Sequential support vector machine classification for small-grain weed species discrimination with special regard to *cirsium arvense* and *galium aparine*. *Comput Electron Agric* 80:89–96
- Salem SB, Bacha K, Chaari A (2012) Support vector machine based decision for mechanical fault condition monitoring in induction motor using an advanced Hilbert-Park transform. *ISA Trans* 51:566–572
- Santana JC (2006) Forecasting time series with neural networks: an application to the Colombian inflation [Predicción de Series Temporales con Redes Neuronales: Una Aplicación a la Inflación Colombiana]. *Revista Colombiana de Estadística* 29:77–92
- Singh A, Imtiyaz M, Isaac RK, Denis DM (2012) Comparison of soil and water assessment tool (SWAT) and multilayer perceptron (MLP) artificial neural network for predicting sediment yield in the Nagwa agricultural watershed in Jharkhand, India. *Agric Water Manag* 104:113–120
- Souhailia S, Bacha K, Chaari A (2012) MLP neural network-based decision for power transformers fault diagnosis using an improved combination of Rogers and Doernenburg ratios DGA. *Int J Electr Power Energy Syst* 43:1346–1353
- Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J (2002) Least squares support vector machines. World Scientific, Singapore
- Trafalis TB, Ince H (2000) Support vector machine for regression and applications to financial forecasting. In: Proceedings of the IEEE International Joint Conference on Neural Networks, Como, 2000, pp 348–353
- Tsai C-F, Wu J-W (2008) Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Syst Appl* 34:2639–2649
- Tung WL, Quek C, Cheng P (2004) GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures. *Neural Netw* 17:567–587
- Üstün B, Melssen WJ, Buydens LMC (2007) Visualisation and interpretation of Support Vector Regression models. *Anal Chim Acta* 595:299–309
- Vapnik V (1995) The nature of statistical learning theory. Springer, Berlin
- Vapnik V (1998) Statistical learning theory. Wiley-Interscience, New York
- Wang H, Fan G (2010) Study on the model of CPI prediction based on BP neural network. In: Proceedings of the 2nd international conference on information technology and computer science, Kiev, pp 573–576
- Wang Y, Wang B, Zhang X (2012) A new application of the support vector regression on the construction of financial conditions index to CPI prediction. *Procedia Comput Sci* 9:1263–1272
- Welfe A (2000) Modeling inflation in Poland. *Econ Model* 17:375–385
- Yang L, Zhang J (2012) Prediction study on anti-slide control of railway vehicle based on RBF neural networks. *Physics Procedia* 25:911–916
- Yeganeh B, Motlagh MSP, Rashidi Y, Kamalan H (2012) Prediction of CO concentrations based on a hybrid Partial Least Square and Support Vector Machine model. *Atmos Environ* 55:357–365
- Yu S, Wei Y-M, Wang K (2012) China's primary energy demands in 2020: predictions from an MPSO-RBF estimation model. *Energy Convers Manag* 61:59–66
- Yümlü S, Gürgen FS, Okay N (2005) A comparison of global, recurrent and smoothed-piecewise neural models for Istanbul stock exchange (ISE) prediction. *Pattern Recognit Lett* 26:2093–2103
- Zeng Z, Zhou J, Tao N, Feng L, Zhang C, Han X (2012) Support vector machines based defect recognition in SonicIR using 2D heat diffusion features. *NDT E Int* 47:116–123
- Zhang J, Sato T, Iai S (2006) Support vector regression for on-line health monitoring of large-scale structures. *Struct Saf* 28:392–406
- Zhong P (2012) Training robust support vector regression with smooth non-convex loss function. *Optim Method Softw* 27:1039–1058