ELSEVIER

Advances in Environmental Research 5 (2001) 405–414

Advances in Environmental Research

www.elsevier.com/locate/aer

Estimating tree mortality of Norway spruce stands with neural networks

Hubert Hasenauer [a,*], Dieter Merkl [b], Martin Weingartner [b]

[a] *Institut für Waldwachstumsforschung, Universität für Bodenkultur, Peter Jordan Straße 82, A-1190 Wien, Austria*
[b] *Institut für Softwaretechnik, Technische Universität Wien, Favoritenstraße 9-11 / 188, A-1040 Wien, Austria*

Accepted 2 January 2001

Abstract

Within forest growth modeling LOGIT models are used to predict individual tree mortality. In this paper we present, Multi-Layer Perceptron, Learning Vector Quantization and Cascade Correlation networks as different formalisms for mortality predictions. The data set for parameterizing the LOGIT model and training the different neural network types comes from the Austrian National Forest Inventory. After training the different network types, we evaluate the resulting mortality predictions using an independent data set from the Litschau forest. The results indicate that Multi-Layer Perceptron with the learning algorithm resilient back-propagation and scaled conjugate gradient and Cascade Correlation with learning algorithm resilient back-propagation perform the best predictions. This suggests that neural networks are a viable alternative to the conventional LOGIT approach.

Keywords: Tree mortality; Neural networks; LOGIT models; Norway spruce

1. Introduction

European forest management has traditionally relied on yield tables to estimate timber production for forest management decisions. The main assumptions concerning yield tables are that the forest stands they describe are pure and even-aged. There is a strong movement toward uneven-aged forest management based on the general understanding that uneven-aged mixed-species stands increase or at least maintain soil fertility, increase biodiversity and improve stand resilience, thereby reducing the susceptibility to physical and/or biotic disturbances. The shift from even-aged forest management renders existing yield tables increasingly unreliable. One possibility to forecast yields for uneven-aged mixed-species stands is to develop stand growth models that operate at the tree level (Wykoff et al., 1982; Pretzsch, 1992; Hasenauer, 1994; Sterba et al., 1995; Pukkala and Miina, 1997). Such individual tree growth models consist of diameter and height increment functions and a mortality model. They have been developed mainly to study stand development in uneven-aged mixed-species stands.

The prediction of tree mortality is one of the most difficult tasks within such growth models. In contrast to yield tables, which simply describe the reduction of the total stem number per unit area over time, tree growth

* Corresponding author. Tel.: +43-1-17654-4205; fax: +43-1-47654-4242.
E-mail address: hasenau@edv1.boku.ac.at (H. Hasenauer).

1093-0191/01/$ - see front matter
PII: S1093-0191(01)00092-2

models must have a mortality model to predict the death or survival of a tree within the projection period (e.g. 5 years). Since Neter and Maynes (1970), this problem is solved with a LOGIT-model according to the following formalism: (1) based on a set of stand and site characteristics the probability of mortality within a given growing period is calculated for every tree within a stand. This results in probabilities for tree mortality ranging between 1 (tree is dead) and 0 (tree is alive); and (2) the final decision whether a tree is dead, and thus has to be removed from the tree list, is done by a random number generator.

Within forest growth modeling, it is customary to predict the probability of mortality as a logistic function. In this work we compare different neural networks architectures, namely Multi-Layer Perceptrons, Cascade Correlation networks and Learning Vector Quantization, with the conventional LOGIT model for Norway spruce as developed by Monserud and Sterba (1999):

$$p = \frac{1}{1 + e^{-(x)}} \tag{1}$$

where p is the probability of mortality and x the vector of the following set of independent variables:

$$x = a_0 + a_1/DBH + a_2 \cdot CR + a_3 \cdot BAL + a_4 \cdot DBH + a_5 \cdot DBH^2 \tag{2}$$

Where, DBH is the diameter at breast height, CR is the crown ratio defined as the relative crown length in relation to the total tree height, BAL is the basal area of the larger trees according to Wykoff (1990) and a_0–a_5, the Maximum Likelihood estimates. For further details on LOGIT models, we refer to Monserud and Sterba (1999), Hasenauer (1994) and Hasenauer (2000).

In our previous work, we introduced neural networks for mortality prediction by comparing the LOGIT model with a self-organizing map (Hasenauer and Merkl, 1997), Learning Vector Quantization (Merkl and Hasenauer, 1998) and some other neural network techniques (Hasenauer and Merkl, 1999). The second of these studies has shown that Learning Vector Quantization produces slightly better mortality prediction than the LOGIT model as well as the self-organizing map. The results of all studies, however, were achieved with only a rather small data and variable set.

The objective of this paper is to extend our work on modeling tree mortality in three dimensions: (1) using the large data set of the Austrian National Forest Inventory which consists of more that 18 000 individual tree observations for Norway spruce all over Austria; (2) test different neural network architectures for their capability to describe tree mortality vs. the classical LOGIT approach. For this step only the input variables given in Eq. (2) are used to train the different neural nets; (3) extend the set of independent variables [vs. Eq. (2)]; and (4) evaluate and interpret all results using an independent data set from the Litschau forest. This step can be viewed as a typical application example for mortality predictions within the framework of an individual tree model.

2. Data

2.1. Austrian National Forest Inventory

The data for this study came from the Austrian National Forest Inventory (ANFI). This inventory is based on a permanent sample plot design systematically distributed all over Austria with a distance of 3.89 km. The permanent plot centers were marked with a hidden iron stake to eliminate research plot bias and to ensure that the forest inventory is representative for growth conditions and forest management throughout Austria. In a given year, every fifth cluster is remeasured. Therefore, there is a representative sample of all Austrian forests in the available data each year.

The permanent sample plots were established from 1981 to 1985. Trees with a diameter at breast height (DBH, 1.3 m) less than 5 cm were not measured. Trees with a diameter at breast height between 5 and 10.4 cm were only measured within a circle of 2.6-m radius from the plot center. Trees with a diameter at breast height larger than 10.4 cm were selected by angle count sampling using a basal area factor of 4 m^2/ha. Sample trees were recorded by their polar co-ordinates.

Every plot was assigned to one of 21 grown districts, to one of 26 soil groups, to one of 20 vegetation types and to one of five soil moisture classes based on the vegetation. For details see Forstliche Bundesversuchsanstalt (1994). The topography is defined by the elevation, the slope and the position of the slope, the soil by soil depth, soil group and thickness of the humus horizons, the vegetation by the vegetation type and the soil moisture. The geographic variable is defined by the growth districts. At plot establishment and plot remeasurement the following data were recorded for every sample tree: species; diameter at breast height (DBH); the total height of the tree (H); and the height to the live crown to determine the crown ratio (CR). As a measure of the social ranking of the tree within the stand, the basal area of the larger trees (BAL) according to Wykoff (1990) was calculated. Competition is determined by the crown competition factor (CCF) (Krajicek et al., 1961) and was calculated for each plot. BAL and CCF describe the competition effects.

Plot descriptors were evaluated within a circle of 300

Table 1
Summary statistics of the training and test data[a]

Variable	ANFI		Litschau	
	Mean	Min–Max	Mean	Min–Max
BAL	20.3	0.0–116.0	23.5	0.0–45.4
DBH (cm)	29.9	5.0–139.0	14.5	5.0–57.2
CR	0.65	0.00–1.00	0.50	0.01–0.95
n_{total}	18671	–	8010	–
n_{dead}	493	–	199	–

[a]Where *BAL* is the basal are of the trees larger in diameter, *DBH* the diameter at breast height, *CR* the crown ratio, n_{total} the number of trees (dead or alive), n_{dead} is the number of dead trees in a given 5-year growth period. The training data come from the Austrian National Forest Inventory (ANFI) and the test data from an independent study at Litschau.

m^2. The elevation is measured to the nearest 100 m and the slope is measured to the nearest 10%. The humus conditions were defined by the thickness of the A, F and H horizons in centimeter. Every tree on a given plot was remeasured after 5 years, between 1986 and 1990. In this study, we will use 18 671 Norway spruce (*Piciea abies* L. Karst) trees (see Table 1).

2.2. The Litschau forest

In 1977 the Institute of Forest Growth Research has designed 22 permanent sample plots within the Seilern-Aspang Forest in Litschau. Litschau is in the northern region of the Waldviertel in Austria. The 22 plots consist of uneven-aged forest stands and have different site conditions. Every sample plot was remeasured in 1982, 1987, 1992 and 1997. 1977–1982 is called growth period 1, 1982–1987 growth period 2, 1987–1992 growth period 3 and 1992–1997 is growth period 4. For our analysis we will use only growth period 3 and 4.

At each measurement the diameter at breast height (*DBH*), the tree height (*H*) and the height to the live crown base to determine the crown ratio (*CR*) was recorded (see Table 1). From these measurements, the basal area of the larger trees (*BAL*) and the crown competition factor (*CCF*) were calculated. Furthermore, according to the instruction of the Austrian National Forest Inventory (ANFI) (Forstliche Bundesversuchsanstalt, 1994) each of the 22 permanent sample plots was assigned to the belonging growth district, the soil group, the vegetation type and to one of the five soil moisture classes. Finally, the thickness of A, F and H horizons was determined.

3. Neural networks: a short tour

The research area of neural networks is a successful field within computer science specialized in providing solutions in application domains that are difficult to model with conventional statistical approaches. Such applications are usually characterized by noisy input data, largely unknown intrinsic structure and changing conditions. The major difference between neural networks and traditional methods of computer science is that the behavior of the former is the result of a training process where typical situations of the application area are presented to the neural network which adapts its structure accordingly. Conversely, in traditional methods of computer science the behavior of the system is predefined, e.g. in form of an algorithm.

Neural networks consist of a number of fairly simple neural processing elements. For simplicity, we will refer to the neural processing elements as units or neurons. The neurons communicate with each other by means of directed and weighted connections. Based on their strategy of training, neural networks are usually divided into models performing: (1) supervised; (2) reinforcement; or (3) unsupervised learning. The distinguishing feature is the amount of information concerning the desired result that is presented to the neural network during learning. The neural networks that are presented in this work are all models adhering to the supervised learning paradigm. In this paradigm, every input data is associated a desired result, i.e. target data. During training, the neural network seeks to adjust its structure in a way that the correct target is produced for each input data.

In the remainder of this section, we provide a brief description of the neural network models that are used during the experiments. For further details, we refer to textbooks on neural network technology (Bishop, 1995; Kohonen, 1995; Müller and Reinhardt, 1991; Zell, 1994).

3.1. Multi-Layer Perceptron

A Multi-Layer Perceptron consists of units and directed, weighted links (connections) between them. In analogy to activation passing in biological neurons, each unit receives an input that is computed from the weighted outputs of units with connections leading to this unit. The inputs to neurons in each layer come exclusively from the outputs of neurons in previous layers, and outputs from these neurons pass exclusively to neurons in following layers.

Multi-Layer Perceptrons are feed-forward networks with one or more layers of units between the input and output units. These additional layers contain the so-called hidden units. These hidden units are not specified by the task and are not part of the input or output layer. The output of each neuron in the network is a function of that neurons input.

The aim of learning is to find a set of weights that

ensure that for each input data the output produced by the network is the same as (or sufficiently close to) the desired output. The learning involves adjusting weights, so that errors, i.e. deviation between computed and target output, will be minimized. The function used to measure errors is the result of the residual sum-of-squared errors.

Perhaps the most widely known training strategy for feed-forward neural networks such as Multi-Layer Perceptron is called back-propagation. This training strategy is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a Multi-Layer Perceptron and the desired output (Rumelhart et al., 1986).

Resilient back-propagation (Rprop) is a local adaptive learning scheme, performing supervised batch learning in Multi-Layer Perceptrons using an adaptive version of the Manhattan-learning rule (Riedmiller and Braun, 1993). The basic principle of Rprop is to eliminate the harmful influence of the size of the partial derivative on the weighting step.

The scaled conjugate gradient-training algorithm is based upon a class of optimization techniques well known in numerical analysis as the conjugate gradient methods. A detailed description of the algorithm can be found in Bishop (1995).

One of the most important aspects of any machine learning paradigm is how it scales according to problem size and complexity. Neural networks like other flexible non-linear estimation methods can suffer from either underfitting or overfitting. A network that is not sufficiently complex can fail to detect the signal in a complicated data set that leads to underfitting. A network that is too complex may fit the noise and not just the signal, which leads to overfitting. Overfitting is critical because it can easily lead to predictions that are far beyond the range of the training data. The complexity of a network is related to both the number of weights and the size of the weights.

The most popular method of regularizing neural networks is called *early stopping*. A part of the training data (i.e. 90%) are used to train the network and the other part (i.e. 10%) are used for the validation set. The basic idea is to control the error of the validation set, i.e. the deviation between computed and desired output when presenting the items of the validation set. As long as the error of the validation set is decreasing, training continues with the data from the training set. If the error of the validation set starts to increase, training is stopped and the neural network configuration yielding the best error on the validation set is used.

The most well known regularize is called *weight decay*. The basic principle is to minimize the sum of squared weights with the error, thus preventing them from growing too large and encouraging redundancy of hidden units.

3.2. Cascade Correlation algorithms

A well-known growing feed-forward network is Fahlman's Cascade Correlation architecture (Fahlman and Lebiere, 1990). This network architecture generates a solution to the given problem by adding hidden nodes one at a time to create a minimal network, minimizing the residual error at each building step. Hence, the Cascade Correlation architecture represents a kind of meta-algorithm, in which classical learning algorithms like back-propagation or Rprop are embedded. Cascade Correlation is characterized as a constructive learning rule. The number of inputs and outputs is defined by the problem.

3.3. Learning Vector Quantization

Learning Vector Quantization (LVQ) represents a family of single-layer neural networks. It consists of a pre-determined number of processing units. Each unit has a d-element reference vector and each unit is associated with one of the classes of the input samples. These units are called codebook vectors. Therefore each class of vector input samples is represented by its own set of codebook vectors. This method is apparently a nearest neighbor method, because the smallest distance of the unknown vector from a set of codebook vectors is sought. The Euclidean distance is used as distance metric.

There are many different learning procedures available for LVQ networks (cf. Kohonen et al., 1995). Each of the procedure moves codebook vectors to try to achieve better classification on the training set by the one-nearest-neighbor rule on the codebook. All training procedures are online algorithms. This means that the examples from the training set are presented in random order, one at a time, and the codebook is updated after each presentation. For our experiments, we relied on the so-called OLVQ1 learning procedure as recommended by Kohonen et al. (1995).

4. Experimental setup

The data from the Austrian National Forest Inventory (ANFI) are used to train the neural networks and for model selection. Every tree in the data set is described by using three features, namely diameter at breast height (DBH), crown ratio (CR) and basal area in larger trees (BAL). The data are linearly transformed to the interval [0, 1] to achieve comparable

values concerning their respective order of magnitude. This kind of data pre-processing is standard for neural network applications.

Our data for training, the ANFI, were randomly split in a training data set and a test data set. These two data sets are used to find a good model complexity. Each method is trained with different model complexity, i.e. number of hidden units or number of codebook vectors, to select the best model complexity. The best model is then trained with the whole ANFI data.

Finally, the trained networks are tested and evaluated with the independent data set from the Litschau forest. This step of our work can be considered as a typical application of mortality models (LOGIT as well as neural networks) within tree modeling.

4.1. Multi-Layer Perceptron (MLP)

In this study, we construct the multi-layer network with one hidden layer. The hidden layer is fully connected to the input and output layers but there is no direct connection between the input and output layers. At the output layer one unit is used. The output 0 means that the tree is alive and 1 that the tree is dead.

The number of hidden units is chosen with model selection. Eighteen thousand six hundred and seventy-one sample trees of the ANFI are used to train the networks, and 8010 observations (measured on 3147 Norway spruce trees) from two 5-year growth periods in Litschau (see Table 1) for testing. The number of hidden units varies between 1 and 30. The network with the best values that minimize the sum-of-squared errors on the test set is chosen. The network is then constructed on all the records in the training set using the chosen number of hidden units. The freely available Neural Network program package SNNS (Zell et al., 1998) is used for training.[1] The following different learning algorithms are used:

- **Back-propagation (MLP − BP)**: for the learning parameter η the value 0.2 was used and early stopping. We achieved the best result with one hidden unit.
- **The resilient back-propagation (MLP − Rprop)**: we used 0.1 for the initial value Δ_0 and 50 for the upper limit Δ_{max}. Weight decay was used, 4 for the weight decay parameter α and early stopping. The best results were achieved with 28 hidden units.
- **The scaled conjugate gradient training algorithm (MLP − SCG)**: The non-critical default values were chosen as suggested in the SNNS manual (Zell et al., 1998). The network was trained until a local minimum. We achieved the best result with four hidden units.

4.2. The Cascade Correlation algorithms (CC)

We selected the SNNS (Zell et al., 1998) implementation of the Cascade Correlation algorithm. No model selection was necessary and a variation of different early stoppings were used. With 90% of the ANFI data the network was trained, and with 10% the validation runs were performed. After adding a new hidden unit and training this new hidden unit, the error of the validation network was recorded. If the new error was higher than the previous one, the training was stopped. Again, the network with the best validation error was selected. Cascade Correlation is defined with CC. The following learning algorithms for Cascade Correlation were performed:

- **Back-propagation (CC − BP)**: the learning parameter η selected was 0.2. The number of epochs were 200.
- **Resilient back-propagation (CC − Rprop)**: the decreasing factor η^+ was 0.5 and the increasing factor η^- 1.2. The number of epochs were 200.

4.3. Learning Vector Quantization (LVQ)

For our analyses the Learning Vector Quantization (LVQ) program package LVQ − PAK (Kohonen et al., 1995) was selected.[2] To initialize the algorithm, the same number of codebook vector was allocated to each class. Training was carried out with the optimized learning rate function (OLVQ 1), a fast and robust LVQ algorithm. The number of iterations was set to 40 times, the number of codebook vectors in OLVQ 1. The number of codebook vectors was chosen during model selection. The number of codebook vectors varied between 10 and 200 units. We achieved the best results with a network setup of 100 units. The overall prediction quality was determined as the ratio between the difference of predicted and observed number of dead trees vs. observed number of living trees. The network was then constructed on all the records in the training set using the chosen number of codebook vectors.

[1] The SNNS program package is available via http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/snns.html.

[2] The LVQ − PAK program package is available via http://www.cis.hut.fi/research/som_lvq_pak.shtml.

5. Experimental results

5.1. Using three input variables

As discussed, the first step of our research was to compare the conventional LOGIT approach with the different neural network models using only the limited set of independent variables given by Eq. (2) (see Monserud and Sterba, 1999). The ANFI data were used for calibration.

Next we applied LOGIT model as well as the trained different neural network types to the Litschau data. In Table 2 we provide the mean mortality predictions of the best neural network setups, i.e. Multi-Layer Perceptron with back-propagation (MLP − BP) and one hidden unit, Multi-Layer Perceptron with resilient back-propagation (MLP − Rprop) and 28 hidden units, Multi-Layer Perceptron with scaled conjugate gradient (MLP − SCG) and four hidden units, Cascade Correlation with back-propagation (CC − BP), Cascade Correlation with resilient back-propagation (CC − Rprop) and Learning Vector Quantization (LVQ − OLVQ1) with 100 Codebook vectors. Additionally, the table contains mortality rates estimated with the conventional LOGIT model developed by Monserud and Sterba (1999). The results reveal the general superiority of the neural network models as compared to the conventional LOGIT approach.

In tree modeling, it is important to evaluate if time trends between observed vs. predicted results exist. Hence, we divided our 22 research plots from Litschau in 5 age classes and calculated the mean relative difference between predicted vs. observed mortality by age class and growth period according to the total number of trees within each group (see Fig. 1). Positive values indicate an overestimation in tree mortality and negative values an underestimation.

Next we were interested, if trends in the range of the input data exist. Such tendencies in the prediction behavior of a model would indicate that a bias is evident. We divided our Litschau data by different basal area (BAL), crown ratio (CR) and diameter at breast height (DBH) classes and calculated the relative difference in the mortality predictions vs. the observed mortality. Fig. 2 gives the mean relative errors by variable class. Positive values result from overestimated mortality rates while negative values indicate an underestimation in the predicted number of dead trees.

5.2. Using more input variables

Figs. 1 and 2 suggest that Multi-Layer Perceptron with the learning algorithm resilient back-propagation (MLP − Rprop) and scaled conjugate gradient (MLP − SCG) and Cascade Correlation with learning algorithm resilient back-propagation (CC − Rprop) performed the best results. Thus, we were interested, if we additionally enhance the resulting neural networks predictions of the three network types by taking advantage of the full set of independent variables provided by the Austrian National Forest Inventory (ANFI). The ANFI data consist of ordinal and nominal variables. The ordinal input variables diameter at breast height (DBH), the total height of the tree (H), the crown ratio (CR), the basal area of the larger trees (BAL), the crown competition factor (CCF), the elevation, the slope, the thickness of the A, F and H horizons of humus were linearly transformed to the interval [0, 1].

The nominal variables are growth district, soil group, vegetation type and soil moisture. Nominal variables were almost always represented in a neural network by using as many input variables as there were values that the variable can take on. Exactly one of these input variables was turned on according to the value of the variable. All of the other input variables from this nominal variable were turned off. This is called *one-of-n*

Table 2
Difference in the 5 year mortality prediction by estimation method (i.e MLP, CC, LVQ and LOGIT with different learning algorithms) and growth period[a]

Total Δ_{dead}	%	Model	Growth period 3		Growth period 4	
			Δ_{dead}	%	Δ_{dead}	%
−6	−0.19	MLP − BP	−44	−2.76	38	2.44
14	0.44	MLP − Rprop	−32	−2.01	46	2.96
16	0.51	MLP − SCG	−33	−2.07	49	3.15
14	0.44	CC − BP	−40	−2.51	54	3.47
6	0.19	CC − Rprop	−38	−2.39	44	2.83
−2	−0.06	LVQ − OLVQ1	22	1.38	−24	−1.54
20	0.63	LOGIT	−32	−2.01	52	3.34

[a] Δ_{dead} gives the difference between predicted and observed number of dead trees and indicates the absolute error, while the relative error [in%] indicates the ratio of Δ_{dead} vs. the observed number of trees. The statistics are based on the independent data set Litschau. Negative values indicate an underestimation and positive an overestimation, respectively.

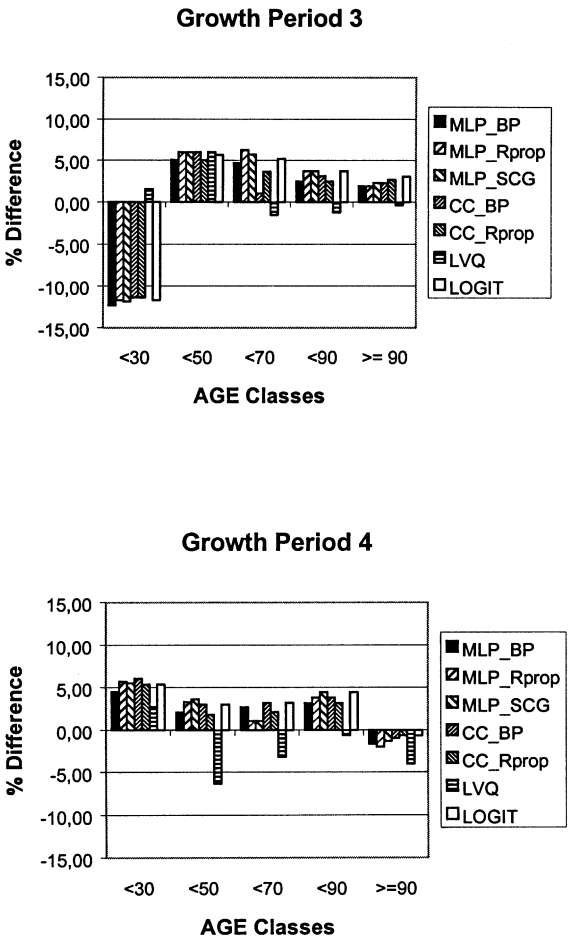

Fig. 1. Relative difference between observed and predicted number of dead trees using: Multi-Layer Perceptron with the learning algorithm back propagation (MLP – BP) resilient back-propagation (MLP – Rprop) and scaled conjugate gradient (MLP – SCG); Cascade Correlation with learning algorithm back propagation (CC – BP) and resilient back-propagation (CC – Rprop); Learning Vector Quantization (LVQ); and the LOGIT model. The statistics show the results for the independent data set in Litschau.

encoding. For the 21 classes of growth districts there were 21 input variables used, for the 26 classes of soil groups 26 variables were used, 20 variables were used for the 20 classes of vegetation types and 5 input variables were used for the five classes of soil moisture. Therefore, the neural network had 82 input variables, 72 as nominal and 10 as ordinal variables, respectively.

The model selection strategy and the parameters for training the neural networks was the same as outlined in Section 4. Again, the independent data from Litschau were used to evaluate the prediction errors. The results are given in Table 3.

A more detailed view of the prediction results based on reduced data set with only 3 independent input variables vs. the approach which takes advantage of the variable set with 82 input parameters is given in Fig. 3. Similar to Fig. 1 we provide the prediction characteristics by age classes and growth period to evaluate for

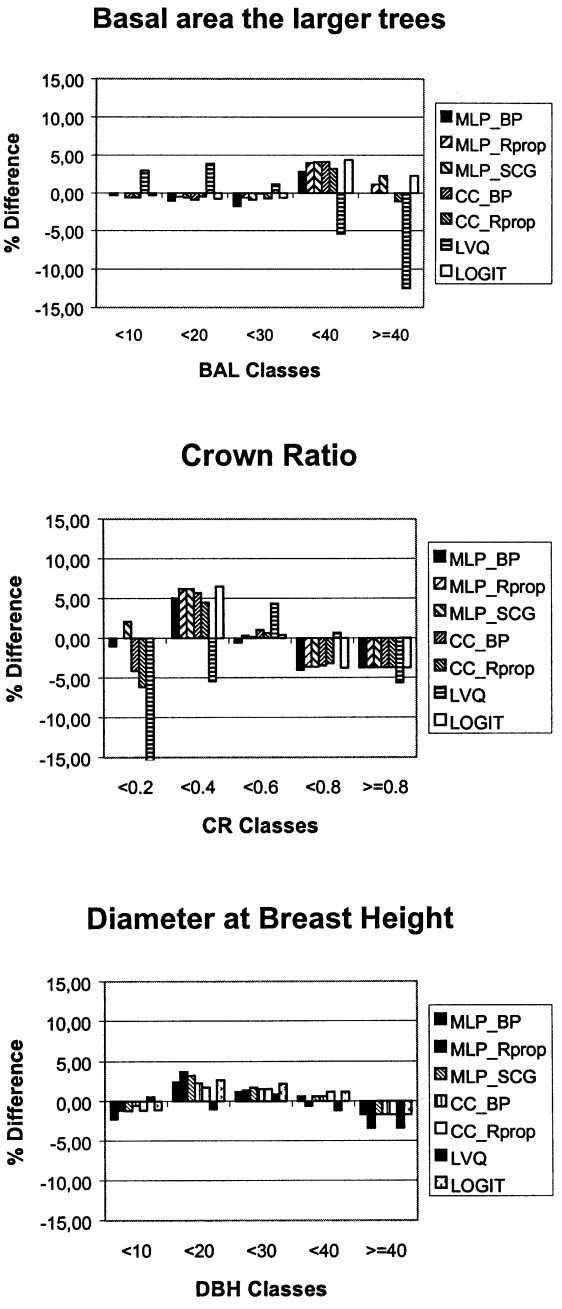

Fig. 2. Relative difference between observed and predicted mortality rates across the actual range of the input values and the different mortality predictions. Negative values indicate an underestimation while positive values indicate an overestimation of actual tree mortality. The statistics show the results for the independent data set in Litschau.

Table 3
Difference in the 5 year mortality predictions by model and growth period (GP3 and GP4) and number of input variables. The figures are based on the independent data set Litschau

Model	3 input variables			82 input variables		
	GP3	GP4	Total	GP3	GP4	Total
MLP – Rprop	−32	46	14	−52	39	−13
MLP – SCG	−33	49	16	−37	27	−10
CC – Rprop	−38	44	6	−47	42	−5

possible time trends in the tree tested neural networks and the LOGIT model. The comparison was based on the relative error in the mortality predictions, i.e. the ratio of the differences between predicted and observed number of dead trees vs. the number of observed trees.

6. Discussion

All tested neural network types performed slightly better mean periodical mortality predictions vs. the conventional LOGIT approach (see Table 2). During our experiments we reached the best estimates (see Fig. 1 and Fig. 2) using a Multi-Layer Perceptron with the learning algorithm resilient back-propagation (MLP – Rprop) and scaled conjugate gradient (MLP – SCG) and Cascade Correlation with learning algorithm resilient back-propagation (CC – Rprop). All three-network types performed a relative mean error rate within the different input variable classes of less than 5% (see Fig. 2). This is an indication that the models are stable and no bias in the resulting predictions is evident.

Most of the models perform an underestimation in the mortality predictions in growth period 3 vs. growth period 4 (see Table 2). Such time pattern are typical for tree mortality within forest stands and is one of the difficult features in modeling because mortality is not only driven by continuous events such as increasing stand density. Very dense stands are labile and water stress, insect attacks or extreme weather conditions may result in periodically higher mortality rates followed by periods of lower mortality.

An interesting result performs the Learning Vector Qunantization (LVQ) networks. From Table 2 and Fig. 1, this network type seems to perform good results, while Fig. 3 indicates that LVQ is more sensitive to certain stand situations as expressed by the relatively high error rates for situations with a basal area (BAL) of > 40 and a crown ratio (CR) of < 0.2. This suggests a certain model bias.

Using all 82 input data of the Austrian National Forest Inventory (ANFI) resulted only in a slight improvement vs. the reduced models (see Table 3 and Fig. 3). Nevertheless, for practical applications a higher model stability can be expected because all available data were used for training and thus the representation of the possible variety of stand and site conditions within Austria is addressed more adequately.

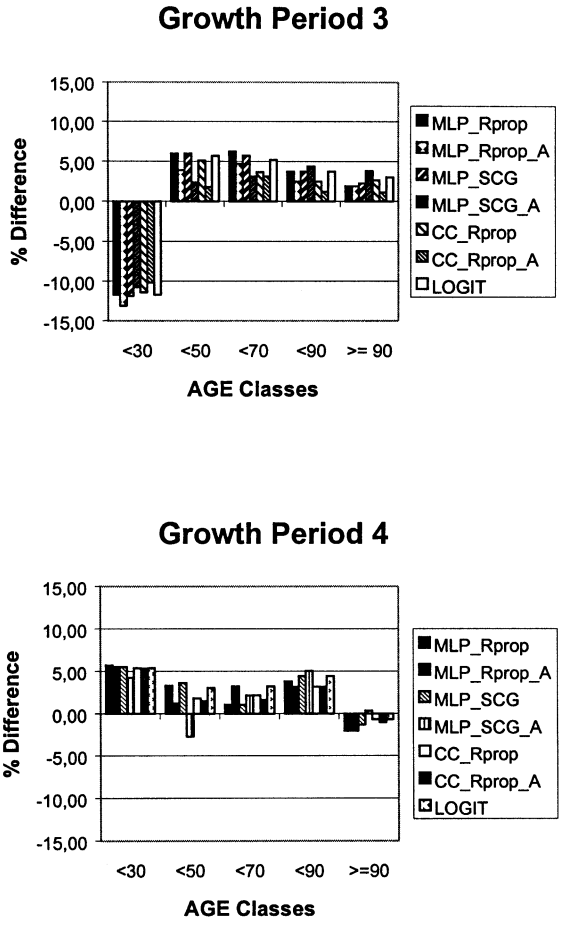

Fig. 3. Relative difference between observed and predicted number of dead trees using: Multi-Layer Perceptron with the learning algorithm resilient back-propagation (MLP – Rprop) and scaled conjugate gradient (MLP – SCG); Cascade Correlation with learning algorithm resilient back-propagation (CC – Rprop); and the LOGIT model. The results with the five independent input variables has no extension and the results using all 82 input variables is labeled with –A. The statistics show the results for the independent data set in Litschau.

An important concern in interpreting trends of over or underestimated mortality rates is that trees which were accidentally considered as dead tress will be eliminated from the tree list while dead trees which were considered as living individuals may have the chance to die during the next growing period. This suggests that the random components whether a tree is dead or alive have a different meaning in a simulation run (Monserud, 1976) because dead trees have no change to become alive in one of the next simulation periods. Thus, tree growth model are usually not that sensitive to a slight underestimation of tree mortality.

7. Conclusion

In our study, we evaluated different types of neural networks to predict tree mortality and compared the results with the conventional LOGIT approach. We used the following neural network architectures: (1) Multi-Layer Perceptron; (2) Cascade Correlation; and (3) Learning Vector Quantization. We were especially interested to determine an optimal network setup in terms of hidden units for the feed-forward network and in terms of codebook vectors for the LVQ network. We divided the data set from the Austrian National Forest Inventory (ANFI) in a learning and a test set, and we used these two data sets to define the complexity of the models. Then we used the whole Austrian National Forest Inventory data for learning.

Our criterion for evaluating the different mortality models was the difference between predicted vs. observed number of dead trees using an independent data set from the Litschau forest. The results indicate that all neural network models perform lower relative mean errors vs. the LOGIT model, an indication that neural network technology may be a viable alternative to the classical LOGIT approach. Additionally, we were able to show that further improvements can be achieved by using more input variables.

For future applications we suggest to investigate other machine learning approaches. Regarding our experiments with more input variables, it is important to note that the effect of using more information to determine the probability of tree mortality has to be investigated in more detail by using other independent data sets similar to the Litschau forest. This would lead to a more thorough understanding of individual tree mortality in uneven-aged, mixed species stands.

Acknowledgements

This work is part of the project ‘Modeling of Eco-Systems with Neural Networks’, sponsored by the Jubiläumsfonds der Oesterreichischen Nationalbank under project number 7058. We thank Karl Schieler and Klemens Schadauer of the Austrian Federal Forest Research Center for making the Austrian National Forest Inventory data available. Finally, we thank the three anonymous referees for their valuable comments.

References

Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Oxford, UK, p. 482.

Fahlman, S.E., Lebiere, C., 1990. The cascade-correlation learning architecture. In: Touretzky, D.S. (Ed.), Advances in Neural Information Processing Systems 2. San Mates, CA, pp. 524–532.

Forstliche Bundesversuchsanstalt, 1994. Instruktionen für die Feldarbeit der Österreichischen Forstinventur. Forstl., Bundesversuchsanstalt Wien, p. 194.

Hasenauer, H., 1994. Ein Einzelbaumwachstumssimulator für ungleichaltrige Fichten-Kiefern- und Buchen-Fichten-mischbestände. Forstl. Schriftenreihe, Univ. f. Bodenkultur, Wien, p. 152.

Hasenauer, H., 2000. Die Simultanen Eigenschaften von Waldwachstumsmodellen. Paul Parey, Berlin, p. 131.

Hasenauer, H., Merkl, D., 1999. Mortalitätsschätzungen in ungleichaltrigen Fichtenwäldern mit Hilfe Neuronaler Netze. In: Rautenstrauch, C., Schenk, M. (Eds.), Proceedings of the 13th International Symposium ‘Informatik für den Umweltschutz’. Verlag Metropolis, Magdeburg, pp. 55–68.

Hasenauer, H., Merkl, D., 1997. Forest tree mortality simulation in uneven-aged stands using connectionist networks. In: Bulsari, A.B., Kallio, S. (Eds.), Proceedings of the International Conference on Engineering Applications of Neural Networks. Stockholm, Sweden, pp. 341–348.

Kohonen, T., 1995. Self-Organizing Maps. Springer-Verlag, Berlin, p. 362.

Kohonen, T., Kangas, T., Laaksonen J., and Torkkola K., 1995. LVQ – PAK: the learning vector quantization program package version 3.1. Laboratory of Computer and Information Science, Helsinki University of Technology.

Krajicek, J.E., Brinkman, K.A., Gingrich, S.F., 1961. Crown competition: a measure of density. Forest Sci. 7, 35–42.

Merkl, D., Hasenauer, H., 1998. Using neural networks to predict individual tree mortality. Proceedings on the International Conference on Engineering Applications of Neural Networks. Gibraltar, UK, pp. 197–204.

Müller, B., Reinhardt, J., 1991. Neural Networks: An Introduction. Springer Verlag, Berlin, p. 266.

Monserud, R.A., 1976. Simulation of forest tree mortality. Forest Sci. 22, 438–444.

Monserud, R.A., Sterba, H., 1999. Modeling individual tree mortality for Austrian forest species. Forest Ecol. Manage. 113, 109–123.

Neter, H., Maynes, S.E., 1970. On the appropriateness of the correlation coefficient with 0.1 dependent variable. J. Am. Stat. Assoc. 65 (350), 501–509.

Pretzsch, H., 1992. Konzeption und konstruktion von wuchsmodellen für Rein- und Mischbestände. Institut für Ertragskunde der Forstl. Forschungsanstalt München. 115, 332.

Pukkala, T., Miina, J., 1997. A method for stochastic multiobjectives optimization of stand management. Forest Ecol. Manage. 98, 189–203.

Riedmiller M., and Braun H., 1993. A direct adaptive method for faster backpropagation learning: the Rprop algorithm. In: Proceedings of the IEEE International Conference on Neural Networks, SA Francisco, CA, pp. 586–591.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Sterba, H., Moser, M., Hasenauer, H., Monserud, R.A., 1995. PROGNAUS ein abstandsunabhängiger Wachstumssimulator für ungleichaltrige Mischbestände. Deutscher Verband Forstl. Forschungsanstalten-Sektion Ertragskunde, Eberswalde/Berlin, pp. 173–183.

Wykoff W.R., Crookston N.L., and Stage A.R., 1982. User's guide to the Stand Prognosis Model, USDA Forest Servive GTR INT-133, p. 112.

Wykoff, W.R., 1990. A basal area increment model for individual conifers in the northern Rocky Mountains. Forest Sci. 36, 1077–1104.

Zell, A., 1994. Simulation Neuronaler Netze. Addison-Wesley, Bonn, p. 624.

Zell, A., G. Mamier M., Vogt N., et al. 1998. SNNS: Stuttgart Neural Network Simulator, User Manual, Version 4.2. p. 346.

Vitae

Hubert Hasenauer is an associate professor at the Institute of Forest Growth Research, University of Agricultural Sciences in Vienna, Austria. He was visiting scientist and guest lecturer at the Virginia Polytechnic Institute and State University, The University of Montana, the Forest Research and Experimental Station in Moscow Idaho and at the University of Ulm in Germany. His research interests are modeling Austrian forest and biogeochemical modeling techniques.

Dieter Merkl holds a position of an associate professor with the Department of Software Technology, Vienna University of Technology. He received his diploma and doctoral degree in social and economic sciences from University of Vienna, Austria, in 1989 and 1995, respectively. His current research interests include neural computation, digital libraries and software engineering.

Martin Weingartner is a Ph.D. student at the Department of Software Technology, Vienna University of Technology. He received his diploma degree in computer science from Vienna University of Technology. His research interests are related to machine learning for real-world applications.