Ecological Modelling 467 (2022) 109932

Contents lists available at ScienceDirect

Ecological Modelling

journal homepage: www.elsevier.com/locate/ecolmodel

Short communications

# The Predicting Tree Growth App: an algorithmic approach to modelling individual tree growth

Juliana G. de S. Magalhaes [a,*], Adam P. Polinko [b], Mariano M. Amoroso [c,e], Gursimran S. Kohli [d], Bruce C. Larson [a]

[a] Department of Forest Resources Management, Faculty of Forestry, University of British Columbia, Forest Sciences Centre, 2424 Main Mall, Vancouver BC, V6T 1Z4, Canada
[b] Department of Forestry, Mississippi State University, Box 9681 Mississippi State, MS 39762, United States
[c] Universidad Nacional de Río Negro, Instituto de Investigaciones en Recursos Naturales, Agroecología y Desarrollo Rural, Mitre 630, San Carlos de Bariloche, Río Negro, Argentina
[d] Department of Computer Engineering, School of Engineering Science, Simon Fraser University, 8888 University Drive, Burnaby BC V5A 1S6, Canada
[e] Consejo Nacional de Investigaciones Científicas y Técnicas, Instituto de Investigaciones en Recursos Naturales, Agroecología y Desarrollo Rural, San Carlos de Bariloche, Río Negro, Argentina

ARTICLE INFO

*Keywords:*
Individual tree growth modelling
Machine learning algorithms
Recurrent neural network
Software

ABSTRACT

PredictingTreeGrowth is free and open-source application software written in Python 3.7 that allows easy and fast development of predictive models using the Recurrent Neural Network (RNN)/Long Short-Term Memory (LSTM) framework. RNNs have an upgraded architecture able to capture tree growth mechanisms related to time ordering and size dependence. The motivation for this App is to demystify the use of Machine Learning algorithms and allow accessibility of Machine Learning algorithms by the scientific community. Its simple graphical user interface (GUI) provides straightforward tools for building predictive models with the RNN algorithm.

## 1. Introduction

### 1.1. A machine learning approach to modelling individual tree growth

The most common approach to modelling individual tree growth is statistical data modelling, such as regression models. Regression models produce a coherent and straightforward picture of predictors and response variables relationship. Nevertheless, tree growth features time ordering and size dependence that these models may not be flexible enough to capture, reducing their predictive performance (Vanclay, 1994). The previous year's weather influences the current year's growth of tree species (Speer, 2010). Previous year's level of carbon dioxide, heat and rainfall determine leaves and roots production, which in the following year, produce new leaves and roots (Pallardy, 2008). Furthermore, adding the right polynomials to account for nonlinear relationship between local site, weather, and competition can be difficult and time-consuming.

Modelling with Machine Learning (ML) algorithms may require some a priori assumption about data format but ML algorithms are flexible enough to learn tree growth while dispensing many statistical assumptions, such as homogeneity of variance, independence of observations and normal distribution of errors (Ramasubramanian & Singh, 2017). One advantage over statistical data modelling is considering predictors and response variables relationship as complex and unknown, rather than a pre-established function (i.e. linear or logistic; Figure 1).. First, ML algorithms explore all functions that can potentially fit an observed training dataset and then find a target function that best maps the predictor-response relationship (Breiman, 2001;Supplementary File 1). Modelling with ML algorithms is a new approach to developing advanced individual tree growth models with superior predictive performance.

Among all ML algorithms currently available, Artificial Neural Network (ANN), is an information processing system that imitates human brain's ability to recognize, associate and generalize patterns (Haykin, 1999). It consists of input, hidden, and output layers of artificial neurons called nodes (Figure 1). Similar to what occurs in a brain, signals of excitation or inhibition propagate through the interconnected nodes. Learning happens by adjusting the strength of connections

* Corresponding author.
*E-mail addresses:* juliana.magalhaes@ubc.ca (J.G.S. Magalhaes), adam.polinko@msstate.edu (A.P. Polinko), mamoroso@unrn.edu.ar (M.M. Amoroso), gkohli@sfu.ca (G.S. Kohli), bruce.larson@ubc.ca (B.C. Larson).

https://doi.org/10.1016/j.ecolmodel.2022.109932
Received 5 October 2021; Received in revised form 20 February 2022; Accepted 2 March 2022

between nodes, known as weights (Smith, 1997). ANN adjustments increase or decrease weight values, and thus the algorithm learns a task without any pre-specified rule.

ANN and some other ML algorithms have already been applied in Forestry research, but their use is limited to specific research designs. For instance, (Wu et al. 2007) described an ANN algorithm that identifies 32 plants by recognizing patterns in leaf structure with accuracy greater than 90% (Thuiller 2003). evaluated the performance of four modelling techniques (Linear, Generalized Additive Models, Regression Tree analysis and Artificial Neural Networks) in predicting the distribution of 61 tree species in Europe, and ANN demonstrate higher area under the curve (AUC) mean value on evaluation data compared to Generalized Linear Models and Generalized Additive Models. (Garzón et al. 2006) compared the performance of ANN and two other ML algorithms (Tree-Based Classification and Random Forest) to forecast the potential areas of distribution of *Pinus sylvestris* as well as of 20 other species throughout the whole Iberian Peninsula, with the resultant map demonstrating a loss of Mediterranean forests (Garzón et al., 2008). Therefore, a knowledge gap remains as to how to utilize ML algorithms to model individual tree growth, and so we framed individual tree growth prediction as a ML task and developed an algorithm that automates modelling.

### 1.2. *Individual tree growth modelling as a computer task: choosing the most appropriate ANN architecture and building models*

Recurrent Neural Network (RNN) is well suited to capture tree growth mechanisms related to time ordering and size dependence. RNN contains a recurrent hidden state whose activation at each time is dependent on that of the previous time (Mandic & Chambers, 2001). The combination of current input ($X_t$) with the previous output ($Y_{t-1}$) is what controls the new output ($Y_t$). The algorithm calculates the error between ANN output and input values with respect to hidden nodes' weights (Figure 1). That error valueis used to adjust weights up or down, depending on which direction it is reduced (Rashid, 2013). Once producing a final output, it is saved and looped back into the network (Gulli & Pal, 2017). The algorithm then calculates the error for each time step and automatically updates the weights. RNN algorithms learn how previous outputs affect new outputs by providing the order of error calculation. The rectified linear unit (ReLU) activation function, defined as $y = \max(0, x)$, accounts for nonlinear relationship between input and output variables (Nwankpa et al., 2018). Final model then has weights setting error to a global minimum.

## 2. Implementation and availability

**PredictingTreeGrowth** is free and open-source simulation software, written in Python 3.7 using Keras application programming interface (Chollet, 2015) and designed to run in Microsoft Windows operating systems. Graphical user interface provides tools for building predictive models with the RNN algorithm (Figure 2). The prototype version (available at https://github.com/jgsmagalhaes/PredictTreeGrowth.git) simulates individual tree growth. The main requirement is observations in a multivariate time series format.

## 3. Use case

We trained individual tree growth models for two endemic trees species of the temperate forests of Argentine Patagonia, *Nothofagus dombeyi* and *Nothofagus pumilio*. The study area encompasses 5.8 hectares of mixed and pure stands growing in the Castaño Overa Valley at the Nahuel Huapi National Park, Rio Negro Province, Argentina (Bianchi et al., 2016). The RNN algorithm learned growth patterns from 61 target trees, 21 *N. dombeyi* and 40 *N. pumilio*. In 2017, we used dendrochronological techniques to reconstruct *N. dombeyi* and *N. pumilio* radial growth patterns. Individual tree annual growth measured basal area increment (BAI), the annual increase in stem area expressed in $mm^2$. The App prototype version allows splitting the observed dataset by year or tree. For this modeling exercise, we divided the BAI series by year with training set of 30 years (1980 – 2009) and testing set of 6 years (2010 – 2015). A 30-year period of training indicated to the RNN algorithm the climatic conditions those species were exposed to, essential information to understanding current and to predict future growth. This is a typical multivariate time series predictive problem that we turned into a machine learning task. The RNN algorithm goal was to find a set of parameters that in year ($t$) and for each individual tree ($j$) could accurately estimate basal area growth as a nonlinear relationship between the effects of local site, weather conditions, and competition. We trained complete (1 to 3) and simple (4 to 6) model types. Within each type, we trained models without competition (1 and 4), with total competition (2 and 5) and with explicit competition indices calculated from each species (3 and 6) (Table 1).

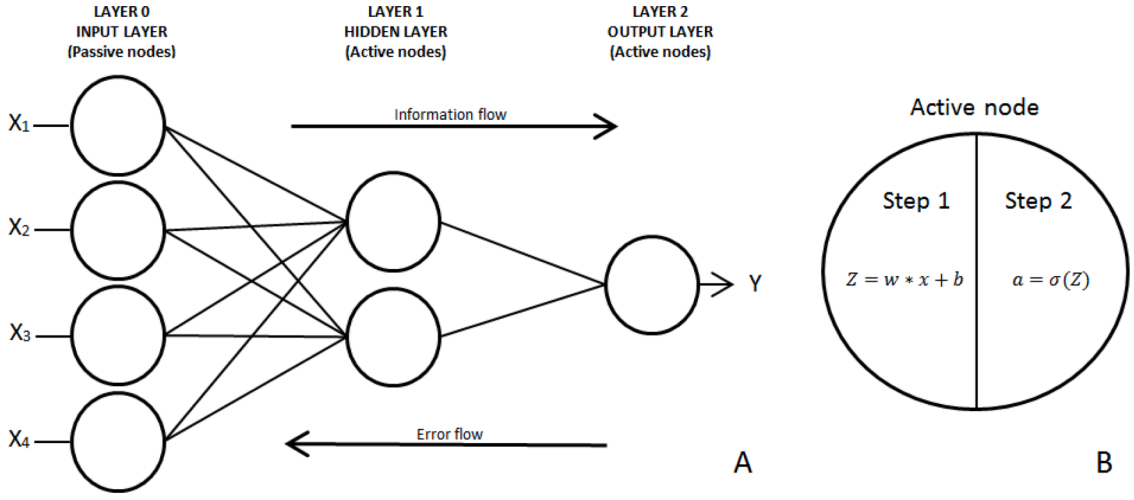


**Fig. 1.** Illustrative representation of the Artificial Neural Network (ANN) learning process. A) ANN architecture encompasses three layers: input, hidden and output. Input nodes feed the network with independent variable values. In the hidden layer, those values are weighted depending on the connection strength between nodes. In this example, all nodes are fully connected to the previous layer's nodes. B) In hidden and output nodes, a two-step computation occurs: first, all input values are weighted ($w$), and a bias ($b$) term is added. After, an activation function ($a$) transforms the sum value of weighted input variables and bias ($Z = w * x + b$) into a limited range. The output node's results are the network's final output. Adapted from Smith (1997).

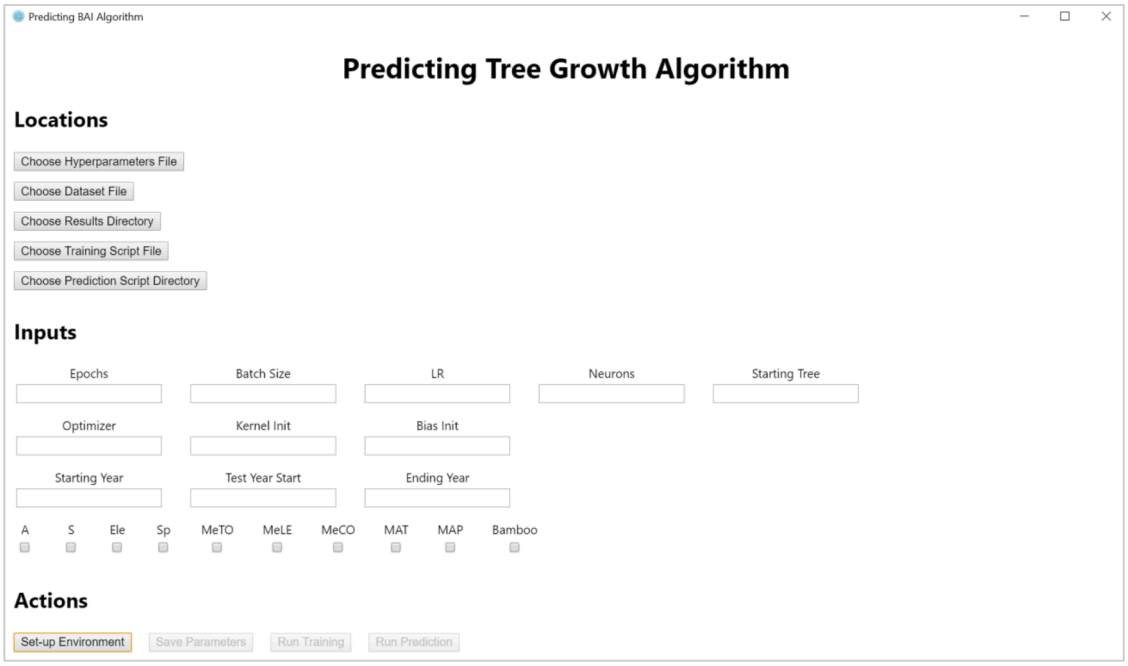


**Fig. 2.** Predicting Tree Growth application software interface showing tools for individual tree growth modeling with the Recurrent Neural Network algorithm. Its interface has three components. 1) Locations loads different hyperparameters, identifies the dataset and output files' path, saves the best model and runs a saved model (*Choose Predictions Script Directory*). Because this is a prototype version, it is possible to test a different script with the button *Choose Training Script File*. 2) Inputs has an area for empirical testing of models. The user can select inputs and hyperparameters controlling the RNN algorithm learning process. 3) Actions is necessary when using the App for the first time to set up the environment in which the RNN algorithm learning process will run. For instance, *Set-up the environment* installs dependencies, such as Keras (Chollet, 2015), SciKit (Pedregosa et al., 2011), Pandas (McKinney, 2011), Numpy (Oliphant, 2006; Van Der Walt, Colbert & Varoquaux, 2011), and Tensorflow (Abadi et al., 2016). Next, the *Run training* trains the algorithm. Once the optimal model is found, weight parameters must be saved (*Save Parameters*), and that model can later be used to make predictions on a new dataset (*Run Prediction*).

**Table 1**
List of models trained in this modeling exercise

| Model | Output (Dependent variable) | Input (Independent variables) |
|---|---|---|
| (1) | BAItj | $= BAI_{(t-1)j} + A + E + S + Bamboo + MAP + MAT$ |
| (2) | $BAI_{tj}$ | $= BAI_{(t-1)j} + A + E + S + Bamboo + CT + MAP + MAT$ |
| (3) | $BAI_{tj}$ | $= BAI_{(t-1)j} + A + E + S + Bamboo + CCO + CLE + MAP + MAT$ |
| (4) | $BAI_{tj}$ | $= BAI_{(t-1)j} + MAP + MAT$ |
| (5) | $BAI_{tj}$ | $= BAI_{(t-1)j} + CT + MAP + MAT$ |
| (6) | $BAI_{tj}$ | $= BAI_{(t-1)j} + CCO + CLE + MAP + MAT$ |

$BAI_{ij}$ is basal area increment in $mm^2$, $t$ is year, $j$ is target tree number, $A$ is aspect in degrees, $E$ is elevation in meters above the sea level (m.a.s.l.), $S$ is slope in degrees, $MAP$ is mean annual precipitation in millimetres, $MAT$ is mean annual temperature in Celsius degrees, $CT$ is total competition index ($CT = CCO + CLE$), $CCO$ is competition from *N. dombeyi* neighbouring trees whereas $CLE$ is from *N. pumilio*. Competition is the distant-dependent index from Martin and Ek (1984):$C_{ij} = \sum_{\substack{i=1 \\ i \neq j}}^{n} \frac{d_i}{d_j} \times e^{-[16 \times dist_{ij}/d_i + d_j]}$, where $n$ is the total number of competitors, $d_i$ is the diameter of competitor $i$, $d_j$ is the diameter of target tree $j$ and $dist_{ij}$ is the distance between competitor $i$ and target tree $j$. The understory of mixed stands is typically of bamboo (*Chusquea* sp.), so we included it as a binary variable (0 = absent and 1 = present) because of its negative influence on development of Nothofagus species in Northern Patagonia (Veblen et al., 1982; Donoso, 1993).

To train all six models, we first indicated dataset file's locations as well as training and testing scripts. After, we specified model architecture according to the grid-search results (batch size of 2, Adam optimizer, learning rate of 0.001 and three neurons in the hidden layer) further explained on Supplementary File 2. We also specified starting years for training (1980) and testing (2009). Finally, we indicated the inputs for each model, and then started the RNN algorithm training process. The algorithm kept species separated and trained the model by adjusting weight parameters to each individual tree. Total computation time was on average of two minutes per model. Example of model configuration output for one individual tree is available on Supplementary File 3.

### 3.1. Visualizing the RNN algorithm learning process

The App gathers training and testing history and tracks the root mean squared error (RMSE), the mean squared error (MSE) and coefficient of correlation (r) metrics. That information is stored in *training_results.csv* file inside *results* folder. That file should generate diagnostic plots, graphical representations of algorithm learning process displaying learning curves, loss function after each epoch, one forward-backward pass of all training samples. In predictive tasks, train and test learning curves' position, shape and trend can indicate two undesirable algorithm behaviours: underfitting and overfitting. Eventually, both behaviours lead to poor model generalization.

### 3.2. Interpreting the RNN algorithm learning process

Trying different hyperparameter configurations is essential in finding a model that fits the training data better and generalizes the testing data well. For this modeling exercise, observed dataset is small and we allowed a long learning process which ultimately favoured an overfitting behavior of the algorithm. Overfitting is depicted in Supplementary File 4 when the RNN algorithm learning process reaches 100 epochs. High epochs imply that training data is excessively passing through the algorithm's computational graph so, it memorized rather than learned relevant patterns in the training data.

Considering that the amount of training data is limited, it was already expected that 100 epochs would make the RNN algorithm overfit. Nevertheless, training with increasing epochs is necessary to help evaluate the most appropriate epoch. It is not possible to visualize a learning improvement after 25 epochs because test and train learning curves are flat. Thus, 50-epoch also caused overfitting. To choose the appropriate epoch, it is essential to observe the metrics displayed in Table 2. Although RMSE values were smaller for models trained with 50 epochs, 30-epoch models featured higher increase in the coefficient of determination ($R^2$) values from training to testing dataset. That increase in $R^2$ values indicated that 30-epoch models generalize better.

### 3.3. Evaluating the quality of RNN individual tree growth model fit

We visually evaluated the quality of all RNN models fit (Supplementary File 5 and 6). Among all 30-epoch trained models, model 6 demonstrated the best fit. Models that included the effects of site and bamboo (Models 1, 2 and 3) were less accurate, even though the algorithm considered those effects to predict BAI. The App applies a fully connected layer, with each hidden neuron connected to every input neuron, and so each connection has a weight. When adjusting weights, the algorithm did not drop any input representing site and bamboo effects, as it usually happens to insignificant variables in statistical data modelling. Since algorithmic modelling considers the training data as true, it is possible that the algorithm recognized aspect, elevation, slope and bamboo, as good predictors of BAI (Breiman, 2001). Models that included the effects of climate (Model 4) as well as weather and competition (Model 5 and 6) were more accurate in predicting BAI, despite an overall overestimating trend as BAI values increased. Usually tree species have higher increment growth with greater precipitation, but that growth differs among individual trees depending on their size and competition (Zhang et al., 2017; Żywiec et al., 2017). Thus, presenting to the algorithm a training dataset with BAI values highly dispersed along the 36-year period might have affected its robustness when predicting growth of large trees.

Based on Model 6 results, representing competition with explicit indices increased model accuracy. In mixed species stands, trees demonstrate intraspecific and interspecific competition. Since species differ in their competitive behavior, both above and belowground, identification of competitors' species is important to make realistic simulations of individual tree growth (Bella, 1971; Zhao et al., 2006; Maleki, Kiviste & Korjus, 2015).

Results from the RNN models did not allow speculation about a negative competition effect on BAI. Higher predictive performance of models 5 and 6 reaffirms necessity of competition and weather inputs to predict BAI. Nevertheless, it is necessary a thorough investigation of weights to conclude about the direction of relationship between inputs (site, weather conditions, and competition) and BAI. For instance, in feedforward networks, the Neural Interpretation Diagram (NID) suggested by Özesmi and Özesmi (1999) illustrates magnitude and mathematical sign of each weight by line thickness and shades, respectively, allowing inferences about individual and interacting inputs' effect on the predicted output. The larger are the weights, the greater is the intensity of signal transfer, and so the more important is that input to predict the output.

The App allowed visualizing the weights estimated during the RNN algorithm process of learning to predict BAI, as listed in Supplementary File 3. However, tracking the weights associated with each input is difficult in fully connected RNNs. It was possible to visualize weights' value and sign for the first hidden layer but after that weights are a combination of previous layer. RNNs contain large number of weights and utilize repeated transformations under time-varying conditions that constrain extracting information about inputs-output relationship with that NID method. Nevertheless, the algorithm considered all types of interactions during the 36-year period of this modeling exercise. Trees' interactions were predominantly beneficial (mutualism) or with one individual having advantage over the others (competition). Even so, the predominant type of interaction assumed in this modeling exercise was competition because changes in weather conditions modify growth resources availability. Therefore, speculations about of *N. dombeyi* and *N. pumilio* under a warmer climate scenario should be based on negative influence that competition has on individual tree growth.

## Conclusion

The App introduces RNN algorithms as new modelling technique, a methodological contribution to current generation of advanced models that help researchers excel in individual tree growth prediction. Lack of big datasets does not constrain its use. Nevertheless, the amount of training data can limit model goodness of fit. The RNN algorithm learns by induction. It can only capture information about tree growth when there is relevant information in the training data. Hence, special attention should be given to the training data, in both quantity and quality, to allow highest predictive performance.

Identification of the optimal RNN tree growth model requires interpretability, and ML models are considered complex systems. However, most of the complexity comes, in fact, from the data rather than the algorithms' lines of code. To improve quality and reliability of ML models generated by the App, it is necessary to understand algorithmic modelling weakness and what reasons induce failing, such as the close relation among bias and variance errors with model complexity.

In sum, adding more parameters and extending the learning process can counterbalance underfitting. Ideally, the optimal RNN model should maximize accuracy of tree growth predictions. It is crucial to provide training samples with relevant information about tree growth mechanisms. Modellers should then strike a balance between high bias error (underfitting) and high variance error (overfitting) to achieve optimal generalization. Although perfect balance of those errors cannot be determined a priori because it depends on the algorithm's task, interpreting algorithm learning process and understanding source of errors can help handle that bias and variance trade-off.

## Author's contribution

JM conceived the ideas and software development; JM and GK implemented the computer code and supporting algorithms; JM led the writing of the manuscript; BL supervised; AP tested the existing code components; AP and MA led the review and editing of manuscript. All authors contributed critically to this draft and gave final approval for publication. All authors declare no conflicts of interest.

## Data accessibility

Not applicable

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ecolmodel.2022.109932.

## References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Kudlur, M., 2016. Tensorflow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16), pp. 265–283. Retrieved from. https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

Bella, I.E., 1971. A new competition model for individual trees. For. Sci. 17 (3), 364–372. https://doi.org/10.1093/forestscience/17.3.364.
Bianchi, E., Villalba, R., Viale, M., Couvreux, F., Marticorena, R., 2016. New precipitation and temperature grids for northern Patagonia: Advances in relation to global climate grids. J. Meteorol. Res. 30, 38–52. https://doi.org/10.1007/s13351-015-5058-y.
Breiman, L., 2001. Statistical modeling: the two cultures. Stat. Sci. 16 (3), 199–231. https://doi.org/10.1214/ss/1009213726.
Chollet, F. (2015). Keras. Retrieved from: https://github.com/fchollet/keras.
Donoso, C., 1993. Bosques Templados de Chile y Argentina: Variación, Estructura y Dinámica. Editorial Universitaria, Santiago, Chile.
Garzón, M.B., Blazek, R., Neteler, M., Dios, R.S.d., Ollero, H.S., Furlanello, C., 2006. Predicting habitat suitability with machine learning models: The potential area of pinus sylvestris L. in the berian peninsula. Ecol. Modell. 197 (3), 383–393. https://doi.org/10.1016/j.ecolmodel.2006.03.015.
E. Garzón, M.B., de Dios, R.S., Ollero, H.S, Bienenstock, S., Doursat, R., 2008. Effects of climate change on the distribution of Iberian tree species Appl. Veg. Sci. 11 (2), 169–178. https://doi.org/10.3170/2008-7-18348Geman. E.Neural Networks and the Bias/Variance DilemmaNeural Computation1992411-58.
Gulli, A., Pal, S., 2017. Deep Learning with Keras. Packt Publishing Ltd.
Haykin, S., 1999. Neural Networks: a Comprehensive Foundation. Prentice-Hall, USA.
Maleki, K., Kiviste, A., Korjus, H., 2015. Analysis of individual tree competition on diameter growth of silver birch in Estonia. For. Syst. 24 (2), 8–21. https://doi.org/10.5424/fs/2015242-05742.
Mandic, D.P., Chambers, J., 2001. Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability. John Wiley & Sons, Inc.
Martin, G.L., Ek, A.R., 1984. A comparison of competition measures and growth models for predicting plantation red pine diameter and height growth. For. Sci. 30 (3), 731–743.
McKinney, W., 2011. Pandas: a foundational Python library for data analysis and statistics. In: Proceedings of the 9th Python for High Performance and Scientific Computing Conference, pp. 51–56. Retrieved from. https://www.dlr.de/sc/portaldata/15/resources/dokumente/pyhpc2011/submissions/pyhpc2011_submission_9.pdf.
Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: comparison of trends in practice and research for deep learning. Retrieved from: https://arxiv.org/abs/1811.03378.
Oliphant, T. E. (2006). A guide to NumPy. Retrieved from http://ns.ael.ru/ports/distfiles/numpybook.pdf.
Özesmi, S.L., Özesmi, U., 1999. An artificial neural network approach to spatial habitat modeling with interspecific interaction. Ecol. Modell. 116 (1), 15–31. https://doi.org/10.1016/S0304-3800(98)00149-5.
Pallardy, S.G., 2008. Physiology of Woody Plants, 3rd ed. Academic Press, Elsevier, San Diego, CA, USA.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Vanderplas, J., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12 (10), 2825–2830. Retrieved from. http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.
Ramasubramanian, K., Singh, A., 2017. Machine Learning using R. Apress, San Bernardino, CA.
Rashid, T., 2013. Recurrent Neural Network Model. Lap Lambert Academic Publishing.
Smith, S.W., 1997. Neural Networks. In the Scientist and Engineer's Guide to Digital Signal Processing. California Technical Publishing.
Speer, J.H., 2010. Fundamentals of Tree-Ring Research. University of Arizona Press, Tucson.
Thuiller, W., 2003. BIOMOD—optimizing predictions of species distributions and projecting potential future shifts under global change. Glob. Change Biol. 9 (10), 1353–1362. https://doi.org/10.1046/j.1365-2486.2003.00666.x.
Vanclay, J., 1994. Modelling forest growth and yield: applications to mixed tropical forests. CAB Int.
Van Der Walt, S., Colbert, S.C., Varoquaux, G., 2011. The NumPy array: a structure for efficient numerical computation. Comput. Sci. Eng. 13 (2), 22–30. https://doi.org/10.1109/MCSE.2011.37.
Veblen, T., 1982. Growth patterns of chusquea bamboos in the understory of Chilean Nothofagus forests and their influences in forest dynamics. Bull. Torrey Bot. Club 109 (4), 474–487. https://doi.org/10.2307/2996488.
Wu, S.G., Bao, F.S., Xu, E.Y., Wang, Y.X., Chang, Y.F., Xiang, Q.L., 2007. A leaf recognition algorithm for plant classification using a probabilistic neural network. In: 2007 IEEE international symposium on signal processing and information technology, pp. 11–16. Retrieved from. https://ieeexplore.ieee.org/abstract/document/4458016.
Zhang, Z., Papaik, M.J., Wang, X., Hao, Z., Ye, J., Lin, F., Yuan, Z., 2017. The effect of tree size, neighborhood competition and environment on tree growth in an old-growth temperate forest. J. Plant Ecol. 10 (6), 970–980. https://doi.org/10.1093/jpe/rtw126.
Zhao, D., Borders, B., Wilson, M., Rathbun, S.L., 2006. Modeling neighborhood effects on the growth and survival of individual trees in a natural temperate species-rich forest. Ecol. Modell. 196 (1-2), 90–102. https://doi.org/10.1016/j.ecolmodel.2006.02.002 https://doi.org/.
Żywiec, M., Muter, E., Zielonka, T., Delibes, M., Calvo, G., Fedriani, J.M., 2017. Long-term effect of temperature and precipitation on radial growth in a threatened thermo-Mediterranean tree population. Trees 31 (2), 491–501. https://doi.org/10.1007/s00468-016-1472-8.

## Further reading

Cherkassky, V., Mulier, F., 2007. Learning from Data: Concepts, Theory, and Methods. John Wiley & Sons.
Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning. Springer New York Inc, New York, NY, USA. Retrieved from. https://web.stanford.edu/~hastie/ElemStatLearn//printings/ESLII_print10.pdfHaykin. S1999Neural Networks: a comprehensive foundation. Prentice-Hall, USA.
Heaton, J., 2008. Introduction to Neural Networks with Java. Heaton Research, Inc.
Kingma, D.P., Ba, J.L., 2015. Adam: a method for stochastic optimization. In: International Conference on Learning Representations, pp. 1–13. Retrieved from. https://arxiv.org/abs/1412.6980.
Michalski, R.S., Carbonell, J.G., Mitchell, T.M., 2013. Machine learning: an Artificial Intelligence Approach. Springer Science & Business Media. https://doi.org/10.1007/978-3-662-12405-5.
Ng, A. (2016, December 31) Advice for applying machine learning: diagnosing bias versus variance [Video file]. Retrieved from https://www.youtube.com/watch?v=fDQkUN9yw44.
Prechelt, L., 2012. Early stopping — but when? In: Orr, G.B., Müller, KR. (Eds.), Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 53–67. https://doi.org/10.1007/978-3-642-35289-8_5.