Intelligent forecasting of growth and development of fruit trees by deep learning recurrent neural networks

To cite this article: A S Kabildjanov *et al* 2023 *IOP Conf. Ser.: Earth Environ. Sci.* **1206** 012015

Intelligent forecasting of growth and development of fruit trees by deep learning recurrent neural networks

A S Kabildjanov, Ch Z Okhunboboeva and S Yo Ismailov*

Department of Automation and control of production processes and production, National Research University Tashkent Institute of Irrigation and Agricultural Mechanization Engineers, 39 Koriy Niyoziy, 100000, Tashkent, Uzbekistan

*E-mail: sarvarismailov.y@gmail.com

Abstract. The questions of intellectual forecasting of dynamic processes of growth and development of fruit trees are considered. The average growth rate of shoots of apple trees of the «Renet Simirenko» variety was predicted. Forecasting was carried out using a deep learning recurrent neural network LSTM in relation to a one-dimensional time series, with which the specified parameter was described. The implementation of the recurrent neural network LSTM was carried out in the MATLAB 2021 environment. When defining the architecture and training of the LSTM recurrent neural network, the Deep Network Designer application was used, which is included in the MATLAB 2021 extensions and allows you to create, visualize, edit and train deep learning networks. The recurrent neural network LSTM was trained using the Adam method. The results obtained in the course of predicting the average growth rate of apple shoots using a trained LSTM recurrent neural network were evaluated by the root-mean-square error RMSE and the loss function LOSS.

1. Introduction

One of the main areas of intensification of horticulture are industries based on the use of «smart» technologies in agriculture. This concept can be implemented in the intelligent system «Smart Garden» through the widespread use of digital technologies and artificial intelligence. The technical basis of such a system should be computer and telecommunications equipment drones and robotic machines. The basis of the mathematical and software of the intellectual system «Smart Garden» are the methods, algorithms and software tools for collecting, systematizing, analyzing, storing and transmitting information [1-4].

One of the tasks of information analysis in the intelligent system «Smart Garden», aimed at effective support for managerial decision-making, is intelligent forecasting of the growth, development and productivity of fruit trees [5, 6]. Intelligent prediction of weakly formalizable dynamic processes, which include the processes of growth, development and ripening of fruits of fruit trees, is currently successfully carried out using recurrent neural networks of deep learning [7]. In this case, the dynamic process is considered as a one-dimensional or multidimensional time series. The most significant results in time series forecasting have been achieved using recurrent neural networks LSTM (Long short-term memory). LSTM networks are designed to solve the problem of long-term dependencies. Memorizing information for a long time is one of the main features of these networks, which does not require long-term training [8, 10].

2. Methods

Development of predictive models for the growth and development of fruit trees. Prognostic models for the growth and development of fruit trees were developed for apple trees of the «Renet Simirenko» variety. The average growth rate of apple tree shoots was predicted. Moreover, the average growth rate of shoots *y* was measured in mm with a frequency of once a day. The measurement period of the parameter *y* was chosen to be 275 days, which corresponds to eight months of a non-leap year, starting from March, during which a change in the growth rate of shoots of fruit trees is observed.

The development of a predictive model for changes in the average growth rate of apple shoots was carried out on the basis of the classical architecture of the LSTM network in the MATLAB 2021 environment using the Deep Network Designer application on a computer with a 12th Gen Intel(R) Core(TM) i9-12900K 3.20 GHz processor and 32.0 GB RAM.

Prediction of the average growth rate of shoots of fruit trees. Loading data with values of the average growth rate of apple shoots $y(t_i); i=\overline{1,N}$ (mm/day), forming a one-dimensional time series, was carried out from a pre-prepared file ‘optim5’. Data download program

```
% Loading Sequence Data
data = load('optim5');
```

After loading the data from the file, a graph of the change in the average growth rate of tree shoots was built, which is shown in Fig. 1. The program for displaying a graph of changes in the average growth rate of shoots

```
% Output of the raw data graph
hPlot=plot(data)
xlabel("Day")
ylabel("mm")
title("The average growth rate of the set of shoots")
grid
set( hPlot, 'LineWidth', 2 );
```

The entire data sample was divided into two parts: training data - 90% of the sequence, testing data - 10%.

```
% Separation into training and test data
%Teaching the first 90% of the sequence
% Test last 10%
numTimeStepsTrain = floor(0.9*numel(data));
dataTrain = data(1:numTimeStepsTrain+1);
dataTest = data(numTimeStepsTrain+1:end);
```

Before training the LSTM network, the data were normalized. Data normalization program

```
% Data normalization
mu = mean(dataTrain);
sig = std(dataTrain);
dataTrainStandardized = (dataTrain - mu) / sig;
```

The data were prepared for the model with a one-dimensional input. To predict future values at each step of the time sequence, responses were specified, which were training sequences with values shifted by one time step. Thus, at each time step of the input sequence, the LSTM network learned to predict the value of the next time step. The predictors were training sequences without the last time step

```
% Preparing a predictor and answers
XTrain = dataTrainStandardized(1:end-1);
YTrain = dataTrainStandardized(2:end);
```

The training data was converted into a data warehouse object. An `arrayDatastore` object was used to convert the predictors and training data responses into ArrayDatastore objects. Combining two data stores was carried out using the `combine` function.

```
arrdsXTrain = arrayDatastore(XTrain);
arrdsYTrain = arrayDatastore(YTrain);
```

```
cdsTrain = combine (arrdsXTrain, arrdsYTrain);
```

The network architecture of the LSTM network was determined, for which the Deep Network Designer application and the command

```
deepNetworkDesigner
```

On figure 1 shows the architecture of a single-layer LSTM network created in Deep Network Designer and the basic settings for layer properties. Changes to the default property settings have been made for sequencelnpuLayer (Name - Input, InputSize - 1, MinLength - 1), lstmLayer (NumHiddenUnits - 200, and fullyConnectedLayer (OutputSize - 1).

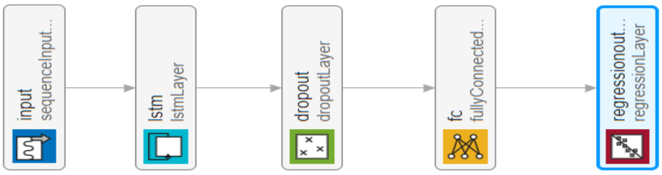
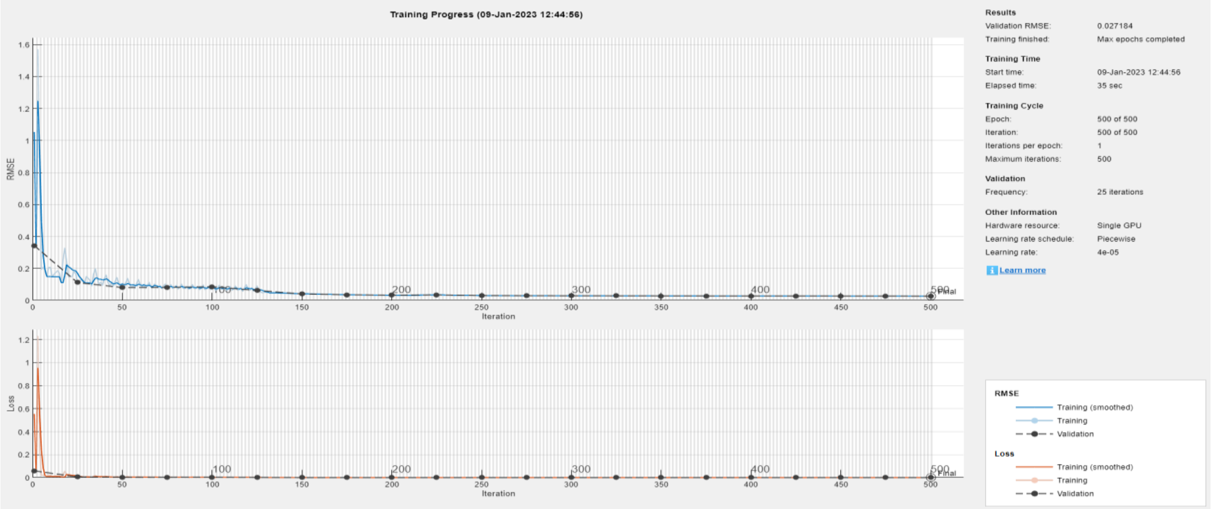


Figure 1. Architecture of a single-layer LSTM network in Deep Network Designer

After the architecture was formed, the resulting network was checked (Analyze option), the results of which determined the absence of errors.

Data was imported from the training data store (Data tab, Import Datastore button). The following values were entered in the opened window: Training data: cdsTrain-CombinedDatastore; Validation data None. The data preview results showed one input time series and one response time series, each with 246 time steps.

The LSTM network was trained using the Adam method. During training, the method parameters were set. To prevent the gradients from exploding, the gradient threshold (`'GradientThreshold'`) was set to 1. The initial learning rate (`'InitialLearnRate'`) was set to 0.005. The learning rate `'LearnRateDropPeriod'`) was reduced after 125 epochs by multiplying by a factor (`'LearnRateDropFactor'`) of 0.2%. The course of training in Deep Network Designer was reflected in the animated graph shown in fig. 2. The trained network was exported to the `Expor` options by the `Export network no workspace` command under the name trainedNetwork_1.



Results
Validation RMSE: 0.027184
Training finished: Max epochs completed

Training Time
Start time: 09-Jan-2023 12:44:56
Elapsed time: 35 sec

Training Cycle
Epoch: 500 of 500
Iteration: 500 of 500
Iterations per epoch: 1
Maximum iterations: 500

Validation
Frequency: 25 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Piecewise
Learning rate: 4e-05

Figure 2. Deep Network Designer LSTM Network Training Progress Graphs

After the network was trained, future time steps were predicted. Before forecasting, test data were normalized

```
dataTestStandardized = (dataTest - mu) / sig;
```

```
  XTest = dataTestStandardized(1:end-1);
  YTest = dataTest(2:end);
```

To initialize the state of the network, a prediction was first made on the `XTrain` training data, then the first prediction was made using the last training response time step `YTrain(end)`.

After that, the remaining predictions were looped and the previous prediction was introduced into `predictAndUpdateState`. To use the CPU for single time step prediction, the `'ExecutionEnvironment'` parameter was set to `'cpu'`.

```
  net = predictAndUpdateState(net,XTrain);
  [net,YPred]=predictAndUpdateState(net,YTrain(end));
  numTimeStepsTest = numel(XTest);
  for i = 2: numTimeStepsTest
  [net, YPred(:,i)] = predictAndUpdateState(net,YPred(:,i-
1),'ExecutionEnvironment','cpu');
  End
```

The forecast standards were changed, which were brought to a normal scale, and the RMSE value was calculated based on non-normalized forecasts

```
  YPred = sig*YPred + mu;
  YTest = dataTest(2:end);
  rmse = sqrt(mean((YPred-YTest).^2))
```

Time series charts were made with predictive values

```
  figure
  plot(dataTrain(1:end-1))
  hold on
  idx = numTimeStepsTrain:(numTimeStepsTrain+numTimeStepsTest);
  plot(idx,[data(numTimeStepsTrain) YPred],'.-')
  hold off
  xlabel("Day ")
  ylabel("mm")
  title("Forecast")
  legend(["Observable" "Forecast"])
```

The predicted values were compared with the test data

```
  figure
  subplot(2,1,1)
  plot(YTest)
  hold on
  plot(YPred,'.-')
  hold off
  legend(["Observable" "Forecast"])
  ylabel("mm")
  title("Forecast")
  subplot(2,1,2)
  stem(YPred - YTest)
  xlabel("Day")
  ylabel("Error")
  title("RMSE = " + rmse)
```

The network state was updated using the observed values of the time steps between predictions by resetting the previous state with the `resetState` command, then initializing the network state and predicting the training data

```
  net = resetState(net);
  net = predictAndUpdateState(net,XTrain);
  YPred = [];
  numTimeStepsTest = numel(XTest);
  for i = 1:numTimeStepsTest
  [net,YPred(:,i)]                                                  =
predictAndUpdateState(net,XTest(:,i),'ExecutionEnvironment','cpu');
  end
```

The obtained forecasts were normalized with previously calculated parameters

```
  YPred = sig*YPred + mu;
  % Calculation of root mean square error
  rmse = sqrt(mean((YPred-YTest).^2))
```

Comparison of forecast values with test data was carried out

```
  figure
  subplot(2,1,1)
  plot(YTest)
  hold on
  plot(YPred,'.-')
  hold off
  legend(["Observable" " Predicted"])
  ylabel("mm")
  title("Forecast with updates")
  subplot(2,1,2)
  stem(YPred - YTest)
  xlabel("Day")
  ylabel("Error")
  title("RMSE = " + rmse)
```

3. Computational experiment

When implementing the prediction of the average growth rate of apple shoots, the selection of the number of memory cells in the LSTM `numHiddenUnits` was carried out by the scanning method. The optimality criterion was the value of RMSE. The complexity of calculations was also estimated, the indicator of which was the time for solving the problem. An illustration of the results of the computational experiment is shown in fig. 3.

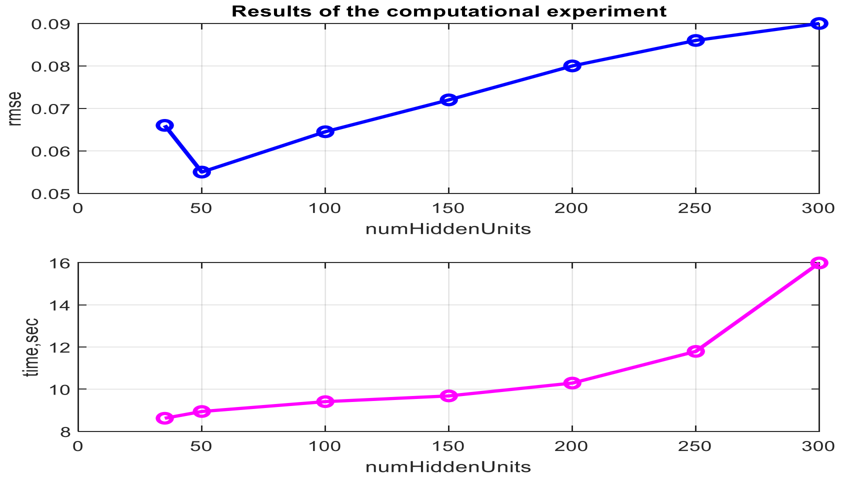


Figure 3. Results of a computational experiment on choice number of memory cells in LSTM `numHiddenUnits`

The final number of memory cells in the LSTM `numHiddenUnits` was chosen to be 50.

It can be seen from Fig. 2 that after 200 epochs, a plateau appears in the change in RMSE and the potential is practically absent. Approximately the same picture was observed with other values of the number of memory cells in the LSTM `numHiddenUnits`, which made it possible to limit the number of epochs to 200. An LSTM network was finally obtained to predict the average growth rate of apple shoots, the results of experimentation with which are shown in Fig. 4 - Fig. 5.

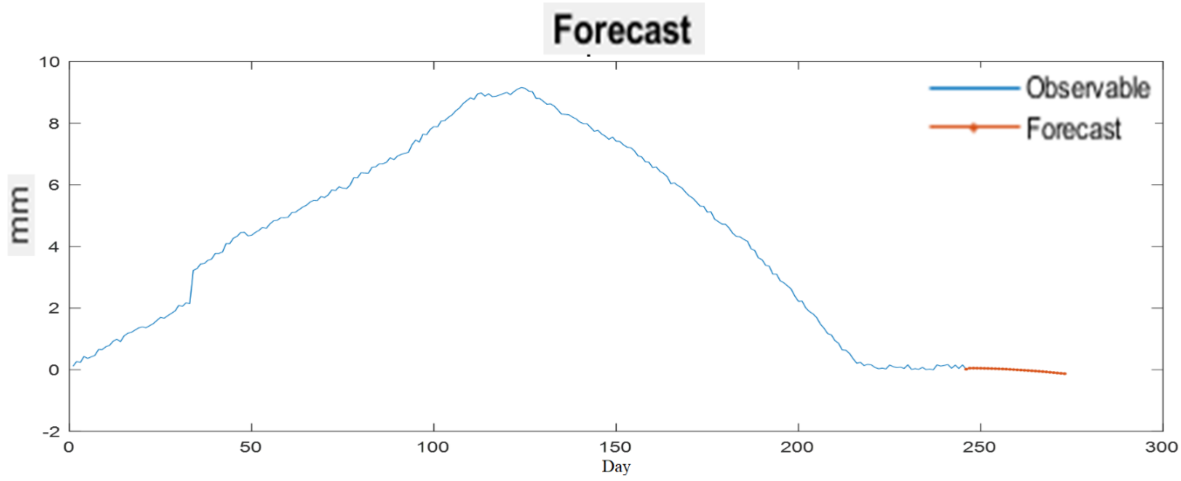


Figure 4. Graph of forecast values

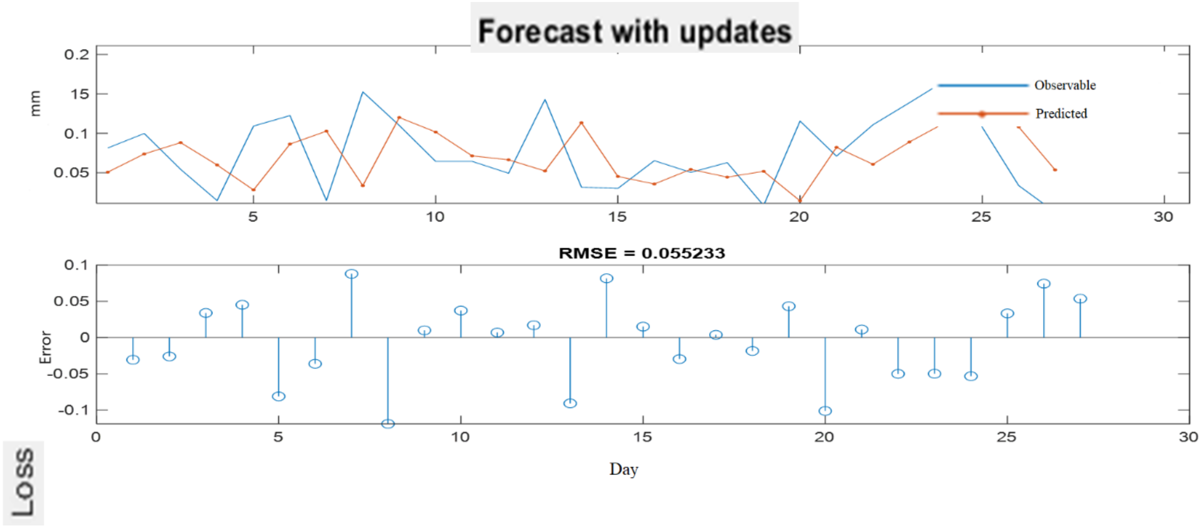


Figure 5. Graphs of initial data and forecasting results, obtained from the trained deep neural network

4. Conclusion

1. The considered technique allows building models for intelligent forecasting of the growth and development of fruit trees in the class of deep learning recurrent neural networks with high prediction accuracy.
2. The above software implementation of the LSTM-network can be widely used in practice for solving problems of intelligent forecasting of difficult-to-formalizable dynamic processes in agriculture.
3. The developed recurrent neural LSTM network for predicting the average growth rate of shoots of apple trees of the «Renet Simirenko» variety can be easily adapted to other fruit trees and used in Smart Garden intelligent control systems.

References

[1] Izmailov A Yu, Smirnov I G, Hort D O 2018 *Digital agricultural technologies in the Smart Garden system* (FGBNU «Federal Scientific Agroengineering Center VIM», Russia, Moscow) p 33-39

[2] Sergienko A V 2008 *Drip irrigation of a young apple orchard on low-growing rootstocks* (dis. ... k.s.-x. Sciences: 06.01.02 Volgograd) 226 p

[3] Ovchinnikov A S, Strekalov S D, Nadvorny A I 2005 *Wave technique in drip irrigation systems* (monograph, Volgograd: Volgograd. state s.-x. acad.) 124 p

[4] Rozhnov S I 2004 *Development of drip irrigation technology for apple seedlings in the conditions of the Lower Volga region* (dis. ... k.s.-x.: 06.01.02, M) 204 p

[5] Artyushin A A, Smirnov I G, Hort D O, Filippov R 2016 Features of the development of an intelligent control system in horticulture *Bulletin of the Michurinsk State Agrarian University* **2** 148-153

[6] Shakirin A I, Lvova O M, Bogdanovich A I 2017 Forecasting the yield of agricultural crops: prospects for the use of artificial neural networks *Processing and quality management of agricultural products: collection of articles of the III International Scientific and Practical Conference* (Minsk: BGATU) p 248-250

[7] *Chollet Francois. Deep learning in Python* (St. Petersburg: Peter, 2018) 400 p

[8] Gers, Schmidhuber. Recurrent Nets that Time and Count 2000 URL: https://neerc.ifmo.ru/wiki/index.php?title=Долгая_краткосрочная_память#cite_ref-LSTM-peephole-connections_6-0K.

[9] Klaus Greff, Rupesh K Srivastava, Jan Koutn´ık, Bas R. Steunebrink, J¨urgen Schmidhuber 2015 LSTM: A Search Space Odyssey URL: https://arxiv.org/pdf/1503.04069.pdf.

[10] Kashirina I L, Demchenko M V 2018 Research and comparative analysis of optimization methods used in training neural networks *Vestnik VSU, Series: system analysis and information technologies* **4** 123-132