Prediction of lung tumor motion extent through artificial neural network (ANN) using tumor size and location data

Biomedical Physics & Engineering Express

RECEIVED
7 November 2015

REVISED
14 March 2016

ACCEPTED FOR PUBLICATION
21 March 2016

PUBLISHED
7 April 2016

PAPER

Prediction of lung tumor motion extent through artificial neural network (ANN) using tumor size and location data

Ines-Ana Jurkovic[1], Sotirios Stathakis[1], Nikos Papanikolaou[1] and Panayiotis Mavroidis[1,2]

[1] Department of Radiation Oncology, University of Texas Health Sciences Center San Antonio, TX, USA
[2] Department of Radiation Oncology, University of North Carolina, Chapel Hill, NC, USA

E-mail: jurkovic@livemail.uthscsa.edu

Keywords: data mining, artificial neural network, 4D CT, motion

Abstract

The aim of this study is to assess the possibility of developing novel predictive models based on data mining algorithms which would provide an automatic tool for the calculation of the extent of lung tumor motion characterized by its known location and size. Data mining is an analytic process designed to explore data in search of regular patterns and relationships between variables. The ultimate goal of data mining is prediction of the future behavior. Artificial neural network (ANN) data-mining algorithm was used to develop an automatic model, which was trained to predict extent of the tumor motion using the data set obtained from the available 4D CT imaging data. The accuracy of the designed neural network was tested by using longer training time, different input values and/or more neurons in its hidden layer. An optimized ANN best fit the training and test datasets with a regression value (R) of 0.97 and mean squared error value of 0.0039 cm^2. The maximum error that was recorded for the best network performance was 0.32 cm in the craniocaudal direction. The overall prediction error was largest in this direction for 70% of the studied cases. In this study, the concepts of neural networks were discussed and an ANN algorithm is proposed to be used with clinical lung tumor information for the prediction of the tumor motion extent. The results of optimized ANN are promising and can be a reliable tool for motion pattern calculation. It is an automated tool, which may assist radiation oncologists in defining the tumor margins needed in lung cancer radiation therapy.

1. Introduction

Currently available treatment planning systems calculate dose distributions on a static CT imaging set. Nevertheless, respiratory induced tumor motion results in noteworthy movement of the tumor volume during the breathing cycle which may lead to considerable discrepancies between the planned and delivered dose distributions (Lujan *et al* 1999, Chui *et al* 2003, George *et al* 2003, Naqvi and D'Souza 2005, Brandner *et al* 2006). This is especially a concern in the lung tumor radiotherapy due to the respiratory induced intra-fraction motion (Cox *et al* 2003). Studies have shown that tumors in the lung can move up to 3–5 cm (Shirato *et al* 2004, 2006). The anatomical motion is somewhat accounted for during treatment planning and delivery by using different methods and approaches (tumor tracking, gating etc) as well as by increasing the 3D margins to the delineated tumor. Therefore, there is a great interest in the development of computational tools that will assist in the tumor margin design, offering reliable ways of reducing them while ensuring that the full extent of the tumor volume is treated.

The generation of a single 4D CT scan involves a digital reconstruction of the CT slices over the respiratory cycle. Published data (Underberg *et al* 2004) have shown that individualized assessment of tumor motion can improve the accuracy of target definition and it is best achieved by using a 4D CT dataset.

This study is inspired by our preliminary findings on the existence of a certain correlation between different factors influencing tumor motion in the lung and the related need to develop a tool that could accurately predict this motion in the 3D space. Our objectives are twofold. First, we want to use the power of the available 4D CT imaging data to predict tumor motion. Until now, a lot of research has been done on this topic (Seppenwoolde *et al* 2002, Liu *et al* 2007, Boldea *et al* 2008, Sonke *et al* 2008). Most of them try to address the respiration-induced tumor motion in the lung by looking specifically at the tumor location and its correlation to the extent of the motion.

Our second objective is the development of a reliable tool by utilizing an artificial neural network (ANN) algorithm. ANNs are proven to be a valuable tool in a multitude of applications (Burke 1994, Bottaci 1997, Wei *et al* 2004, Dogra *et al* 2013, Ahmad *et al* 2014, Kourou *et al* 2015), as well as in cancer diagnosis and prediction. In the latter case, the network is designed to look at pattern recognition and use it to translate input (e.g. different biopsy attributes) and output (cancer categories) data. The potential usefulness of this tool with the right set of data is significant (Konstantina Kourou 2014). Neural networks are trained using different samples and are used in numerous applications; for instance in bioinformatics, optical character recognition, object detection, image processing, stock market prediction, modeling human behavior, loan risk analysis, pattern classification, cancer prognostics (Burke 1994, Bottaci *et al* 1997, Wei *et al* 2004, Dogra *et al* 2013, Ahmad *et al* 2014, Konstantina Kourou 2014).

In this study, neural networks are used for tumor motion prediction based on the different factors that may be influencing its motion extent, as they were identified in our preliminary work.

2. Material and methods

2.1. Dataset

The current study was performed on the 4D CT datasets of 11 radiotherapy patients who were treated for lung cancer. The dataset of each patient included 12 subsets: maximum intensity projection (MIP), average intensity projection (AIP) and ten equally spaced phases of equal duration (the breathing cycle was sampled at ten different instances and a CT dataset was created for each instance). The clinical target volume was delineated on each set of images for each patient.

The reference dataset for the tumor motion assessment was the AIP CT. After performing a DICOM registration for aligning each CT set with the average CT dataset, the corresponding tumor volumes were copied to the AIP CT, where motion was subsequently determined by looking at their related center of the mass. The same procedure was applied on the MIP CT dataset.

2.2. Artificial neural network

In 1943, Warren S McCulloch, a neuroscientist, and Walter Pitts, a logician, developed the first theoretical model of an ANN. In their paper, ‘A logical calculus of the ideas imminent in nervous activity’, they describe the concept of a neuron, a single cell living in a network of cells that receives inputs, processes those inputs, and generates an output (Shiffman 2012). ANN is inspired by the structure of the brain and consists of a set of highly interconnected entities, called neurons. Each accepts a weighted set of inputs and responds with an output, figure 1. ANN has been applied in clustering, pattern recognition, object detection, image processing, function approximation, and prediction systems. ANN uses several architectures and can be trained to solve problems by using a teaching method and sample data. If proper training is put in place ANN has the ability to recognize similarities among different input data. As such, it represented an ideal tool that can utilize similarities found in tumor size and/or location.

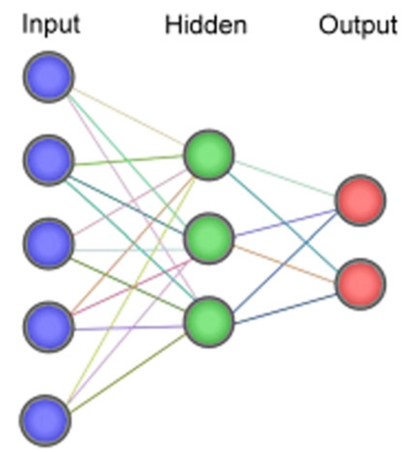

Figure 1. An ANN layer of neurons.

One of the most commonly used ANNs is the multilayer perceptron (MLP). In machine learning, perceptron is a type of linear classifier, i.e. an algorithm for supervised classification of an input into one of several possible outputs. MLP maps set input data onto a set of appropriate outputs and it consists of multiple layers of nodes with each layer being connected to the next one. Apart from the input nodes, each node is a neuron. MLP utilizes the backpropagation (BP) algorithm for supervised network training. BP algorithm is capable of handling large learning problems as it looks for the minimum of the error function in weight space using the method of gradient descent. There are different variants of this algorithm: quasi-Newton, conjugate gradient BP, one-secant BP, Levenberg Marquardt (LM), resilient BP, and many others (Møller 1993). The designed ANN was programmed using the neural network toolbox in MATLAB (Beale *et al* 2010). In MATLAB training and learning functions are mathematical procedures used to automatically adjust the network’s weights and biases. In MATLAB’s perceptron, networks initial values of the weights and biases are zeros.

In this work, the LM algorithm was adopted to train the designed ANN. The LM algorithm is a standard method used to solve nonlinear least squares problems. Least squares problems arise when fitting a parameterized function to a set of measured data points by minimizing the sum of the squares of the errors (SSE) between the data points and the function. Nonlinear least squares problems arise when the function is not linear in the parameters. Nonlinear least squares methods involve an iterative improvement to parameter values in order to reduce the SSE between the function and the measured data points. The LM algorithm interpolates between the following two minimization algorithms: the gradient descent method and the Gauss–Newton algorithm. In the gradient descent method, the SSE is reduced by updating the parameters in the direction of the greatest reduction of the least squares objective. In the Gauss–Newton algorithm, the SSE is reduced by assuming the least squares function is locally quadratic, and finding the minimum of the quadratic. The LM algorithm is more robust than the Gauss–Newton algorithm, and is a very popular curve fitting algorithm used in many software applications (Gavin 2011). The main advantage of this algorithm is that it requires less number of data for training and achieves accurate results. The network training function updates weight and bias values according to LM optimization.

The default network for function fitting problems, a feed forward network, with the default tan-sigmoid transfer function in the hidden layer and linear transfer function in the output layer was used as this type of network can be used for any kind of input to output mapping (Beale *et al* 2010). A feedforward network with one hidden layer and enough neurons in the hidden layer can fit any finite input–output mapping problem. The network was created with varying number of neurons in the input layer and three neurons in the output layer, because there are three target values associated with each input vector. In our network design, once the training started, the number of neurons needed in the hidden layer was determined experimentally as there is no precise rule for their selection (Othman and Ghorbel 2014). More neurons require more computation, and they have a tendency to over fit the data when the number is set too high, but they allow the network to solve more complicated problems. The error criterion that was used for training is mean square error (MSE), as training functions used utilizes the Jacobian for calculations, which assumes that performance is a mean or sum of squared errors (SSE). Therefore, networks trained with this function must use either the MSE or SSE performance function and MSE is the default performance function for feed forward network. In the designed network, MSE is the average squared difference between output and target values. The network algorithm adjusts the weights and biases of the network so as to minimize this MSE. Another default measure used to validate the network performance is regression; *R*. *R* values measure the correlation between output and target values. An *R* value of 1 means a close relationship (Beale *et al* 2010).

ANN training algorithms seek to minimize error in neural networks, however local minima can be a problem, and this problem can be addressed by varying the number of neurons in the hidden layer up until the acceptable accuracy is achieved. The first step was to find the suitable number of neurons in the hidden layer of the ANN architecture.

The number of iterations (called epochs) was set to stop the training when the best generalization was reached. This was achieved by partitioning the data into different sub datasets: training, validation and testing. In the ANN training, a set is used to train the network while another validation set is used to measure the error; network training stops when the error is increasing for the validation dataset. Values belonging in each subset are randomly chosen and they change on each training step (Beale *et al* 2010). The values used in a testing set have no effect on training and so provide an independent measure of the network performance during and after training. The number of neurons in the hidden layer was varied from 1 to 30 and the MSE and regression values for each trial were recorded. Each time a neural network is trained, it can result in a different solution due to different initial weight and bias values and different divisions of data into training, validation, and test sets. Weight and bias values are automatically updated according to the LM optimization. Network default values were used for training parameters such as maximum number of epochs, performance goal, maximum validation failures, initial adaptive learning parameter value, etc.

Other parameters such as subset partition and number of variables in the input dataset were varied through training process and their MSE and regression values were recorded in order to determine which of the available parameters influence output, i.e. accuracy of the prediction, the most. The trained network was then used to predict motion extent for the various sets of the tumor attributes.

2.3. ANN input dataset

The dataset used as the input was extracted from our preliminary study (Jurkovic *et al* 2016). That study compiled tumor motion and volume change data based on the tumor size and/or location and it investigated their relationship patterns. In this study for each of the studied patients the whole respiratory motion through the phases was divided in the two separate trajectories—inhale and exhale. Similarity between the inhalation and exhalation trajectories was evaluated by using two different measures, dynamic time warping and Fréchet distance, and the assessment essentially showed that motion trajectories in the inhale and exhale phases do depend on the location but also depend significantly on the tumor size, i.e. it was shown that two categories of patients have most similar inhale–exhale trajectories: patients with larger tumor volumes (>100 cm^3) regardless of the tumor

Table 1. Input dataset.

Patient #	Tumor size (cm^3)	Lung	Lobe	Location 1	Location 2
1	1.66	Right	Middle	Anterior	Central
2	11.04	Right	Lower	Posterior	Peripheral
3	21.90	Right	Upper	Anterior	Central
4	2.48	Right	Upper	Posterior	Peripheral
5	108.93	Left	Upper	Posterior	Central
6	14.11	Left	Lower	Posterior	Central
7	29.91	Left	Upper	Posterior	Central
8	21.70	Left	Lower	Posterior	Central
9	96.55	Right	Upper	Anterior	Central
10	23.47	Right	Lower	Posterior	Central
11	17.75	Right	Lower	Anterior	Central

location, and patients with tumor volumes >15 cm^3 that are located in the upper lung lobe. Regression analysis was also performed when comparing the similarity between the motion paths through the whole breathing cycle among the different patients versus the tumor size and location in order to establish if there is any correlation. Calculated coefficient (0.73) indicated a moderate linear relationship between these variables. This information was subsequently applied towards the network design and input data extraction. The variables used were tumor volume as delineated on the AIP CT dataset and tumor location in the lung: left/right lung, upper/middle/lower lobe, anterior/posterior location, and/or central/peripheral location, table 1.

Different subsets of the input dataset were used to establish the best training dataset for the designed network and algorithm.

2.4. ANN output dataset

Regarding the output dataset that was used for training, the validation and testing were based on the findings of the preliminary study that calculated the maximum motion extents in the superior–inferior (SI), left–right (LR) and anterior–posterior (AP) directions (table 2). The motion extents were calculated using the tumor volume center of mass (COM) positions through the ten breathing phases in relation to the COM of the tumor volume on the AIP CT dataset. This position was used as a coordinate system center with coordinates 0,0,0.

Based on the foregoing analytical decisions, ANN was designed in MATLAB and one of the resulting network architectures is shown in figure 2. This is a feed-forward network that has one-way connections from input to output layers and it is one of the four available types of supervised neural networks (Beale *et al* 2010). It is most commonly used for prediction, pattern recognition, and nonlinear function fitting.

3. Results

The best generalization was achieved by partitioning the data into 80% training, 10% validation and 10% testing sub datasets. Several different ANN configurations were initially tested with varying number of neurons in the hidden layer. The result with the best MSE and regression values was then used to determine the appropriate number of neurons in the hidden layer of the ANN. We noticed that with increasing number of neurons beyond 10 we did not get any improvement in MSE and R values.

Table 2. Output dataset (maximum motion extent).

Patient #	LR (cm)	SI (cm)	AP (cm)
1	0.19	0.62	0.45
2	0.25	1.00	0.38
3	0.73	0.25	0.24
4	0.20	0.75	0.34
5	0.06	0.25	0.21
6	0.15	0.37	0.25
7	0.25	0.37	0.28
8	0.10	0.76	0.15
9	0.25	0.37	0.25
10	0.28	1.25	0.25
11	0.27	0.75	0.45

Prediction error was calculated for each of the trials according to the following: error = desired output − guessed output.

Subsequently, we further examined the network by varying the number of variables in the input dataset. Again, the best result was achieved when all the available variables were taken into account. However, even when the number of variables was lowered down to two (looking only at the tumor size and location in the lung lobes: lower, middle, upper to predict motion extent), the prediction gave a maximum MSE value of 0.0099 cm^2 with a regression value of 0.93. Compared to our output data this is translated to a maximum difference of −0.31 cm of the motion extent in SI direction between the target data and data predicted with ANN (predicted value was higher than the measured one). The corresponding error in the LR direction was −0.17 cm and in the AP direction it was 0.15 cm (table 4). To clarify this point, since MSE is the average squared difference between all the output and target values and R is regression value that measures correlation between all the output and target values, the

Table 3. The ANN experimentation results for varying numbers of the partition subsets, input factors, and neurons in the hidden layer.

ANN	Partition subsets	Hidden layer neurons	Input data	MSE (cm^2)	Regression
1	70%, 15%, 15%	30	5	0.0391	0.846 91
2	50%, 25%, 25%	10	5	0.0207	0.849 80
3	80%, 10%, 10%	10	5	0.0039	0.971 71
4	80%, 10%, 10%	30	5	0.0051	0.963 26
5	80%, 10%, 10%	30	3	0.0335	0.740 57
6	80%, 10%, 10%	10	2	0.0099	0.928 57
7	70%, 15%, 15%	10	5	0.0108	0.932 88
8	70%, 15%, 15%	1	5	0.0241	0.827 34
9	70%, 15%, 15%	3	5	0.0223	0.843 59
10	70%, 15%, 15%	6	5	0.0106	0.927 72
11	80%, 10%, 10%	10	4	0.0071	0.954 33
12	80%, 10%, 10%	10	3	0.0067	0.951 74
13	80%, 10%, 10%	30	3	8.97E-05	0.999 17

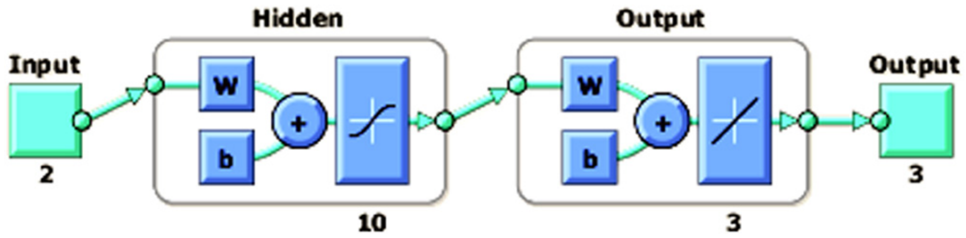

Figure 2. Example of the MATLAB ANN architecture.

Table 4. Motion extent error values for the ANN with the two input parameters.

Direction/patient #	1	2	3	4	5	6	7	8	9	10	11
LR (cm)	0.00	0.08	0.02	−0.02	0.00	−0.06	−0.01	−0.17	0.00	−0.01	0.03
SI (cm)	0.00	0.29	0.02	−0.03	0.00	−0.31	−0.01	−0.24	0.00	0.04	0.01
AP (cm)	0.00	0.05	−0.01	0.00	0.00	−0.07	0.01	−0.11	0.00	0.01	0.15

maximum differences (errors) between output and target values for each motion direction for a given network's MSE and R values can be calculated based on the following equations, (Beale *et al* 2010):

$$\{p_1, t_1\}, \{p_2, t_2\}, \ldots, \{p_Q, t_Q\},$$

$$\mathrm{MSE} = \frac{1}{Q}\sum_{k=1}^{Q}(t(k) - n(k))^2,$$

where p_Q is an input to the network, t_Q is the corresponding target output, and n_Q is the network output. The default regression equation between inputs and outputs is a curve in three-dimensional input space. However, the plots and total regression values reported are the one-dimensional regressions of output versus target:

$$y = bx + a,$$

where y is the output and x is the target value, and R is the correlation between x and y.

Furthermore, to also test the ANN we varied number of subsets that were used for training, validation and testing. The network performed best when more data was used for training. For each run, the MSEs and regression values were recorded. The lowest error and best regression values, looking at the network that also used maximum number of input variables, were achieved with ANN#3, MSE $= 0.0039\ \mathrm{cm}^2$ and regression value $= 0.971\,71$ (table 3). This table actually shows the ANN runs that gave the best results with the different setups after retraining was performed.

In ANN, an epoch (cycle) is a measure of the number of times that all the training vectors are used to update the weights (network is presented with a new input pattern). It is the number of iterations needed to achieve adequate network training, i.e. until minimum gradient is reached. In our case, depending on the number of input variables, the values in the partition sets and the number of neurons in the hidden layer, final number of epochs during training ranged from 5 to 50 (for the runs with the lower number of the input parameters). An example of the network performance is shown in figure 3. This figure shows when best validation performance is reached. Epoch 6 in this example indicates the iteration at which the validation performance reached a minimum. And then the training continued for a few more iterations before the

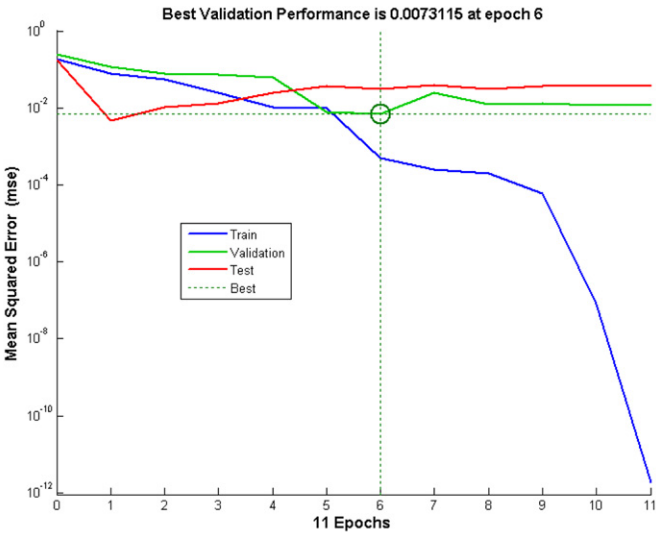

Figure 3. ANN training performance example (with epoch 11 minimum gradient reached).

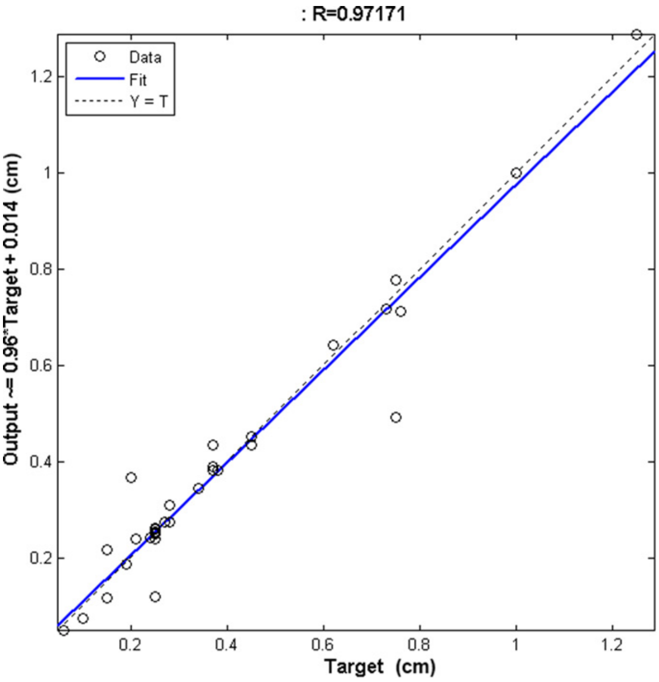

Figure 4. Regression plot for the best ANN with 10 neurons in the hidden layer.

training stopped. The figure does not indicate any major problems with the training. The validation and test curves are very similar. If the test curve had increased significantly before the validation curve increased, then it is possible that some overfitting might have occurred. Generally, the error reduces after more epochs of training, but might start to increase on the validation data set as the network starts overfitting the training data. In the MATLAB's Neural Network Toolbox default setup, the training stops after six consecutive increases in validation error, and the best performance is taken from the epoch with the lowest validation error (Beale *et al* 2010).

The regression plot and error histogram for the best performance run are shown in figures 4–6, ANN #3, table 3. These figures present examples of the plots available and used as a tool to analyze neural network performance after training. In figure 4, the dashed line represents the perfect result—outputs = targets. The solid line represents the best fit linear regression line between outputs and targets. The *R* value is an indication of the relationship between the outputs and targets. In this case, a good fit is indicated for data results that show *R* values greater than 0.9. The scatter plot shows that certain data points have poor fits. The next step would be to investigate these data points and

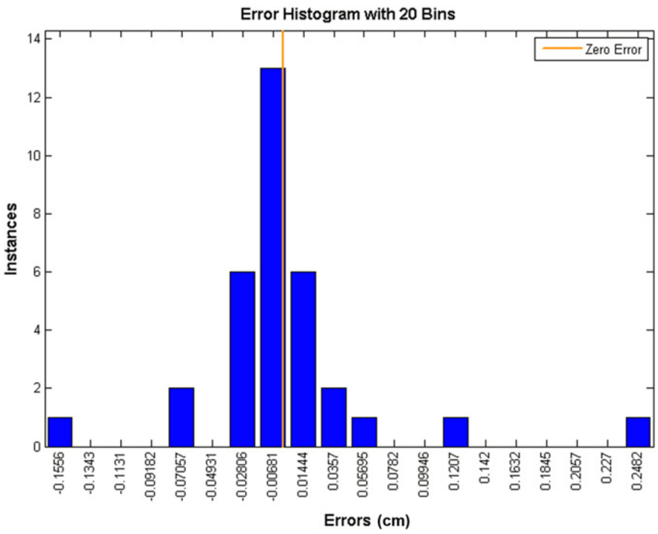

Figure 5. Error histogram plot for the best ANN with 10 neurons in the hidden layer.

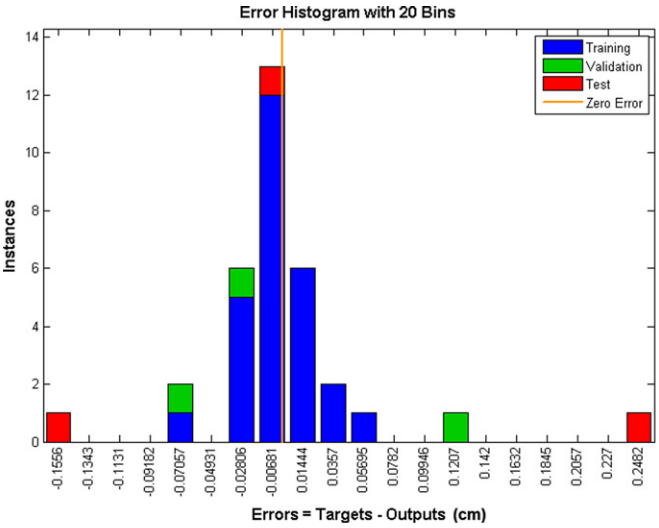

Figure 6. Error histogram plot for the best ANN with 10 neurons in the hidden layer, with the partitions also shown.

determine if they should be included in the training set, in which case we would need additional data in the test data set. Error histograms plotted in figures 5 and 6 provide additional verification of network performance. In figure 6, the blue bars represent training data, the green bars represent validation data, and the red bars represent testing data. This histogram can be used to point out outliers, which are data points where the fit is significantly worse than the majority of data. In this case, we notice that while most errors fall between −0.07 and 0.06, there is a validation point with an error of 0.12 and test points with errors of −0.16 and 0.25. These outliers are also visible on the regression plot, figure 4. We can check the outliers to determine if the data is bad, or if those data points are different than the rest of the data set. If the outliers are valid data points, but are unlike the rest of the data, then the network is extrapolating for these points. Next step would be to collect more data that looks like the outlier points, and retrain the network, i.e. have a larger data set for training and testing of a neural network. Same reasoning applies to a data plotted in the figures 7 through 9. As shown in table 3, the 5th ANN had three input variables, which were chosen for the network design (tumor size, upper/lower lung and anterior/posterior location). The 6th ANN had two input variables chosen for the network design (tumor size and upper/lower lung location). Finally, the 11th ANN had four input variables chosen for the network design (tumor size, left/right lung, upper/lower lung, and peripheral/central tumor location). In all the cases, when the regression values for all the subsets were >0.8 and the MSE values <0.005 cm^2, retraining would stop. This was achievable for all the ANNs except for the one that had 30 neurons in the hidden layer and three input variables (number 5). However,

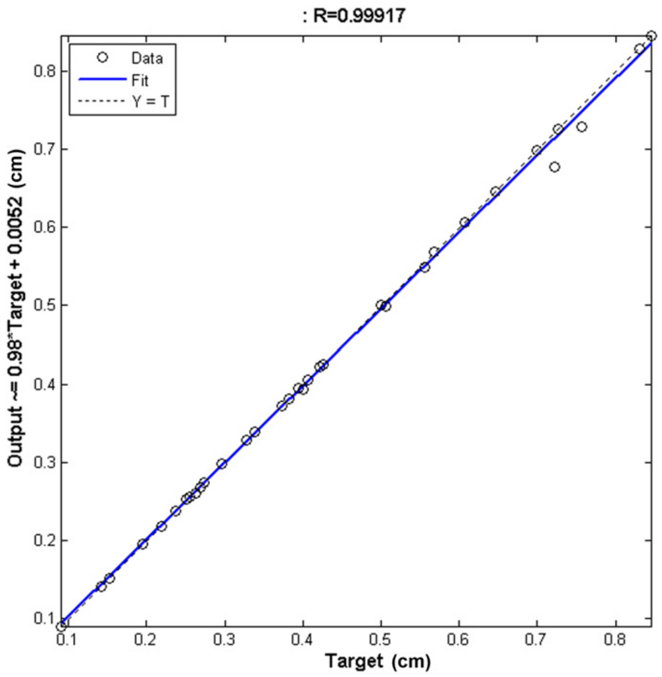

Figure 7. Regression plot for the best ANN with 30 neurons in the hidden layer after excessive retraining.

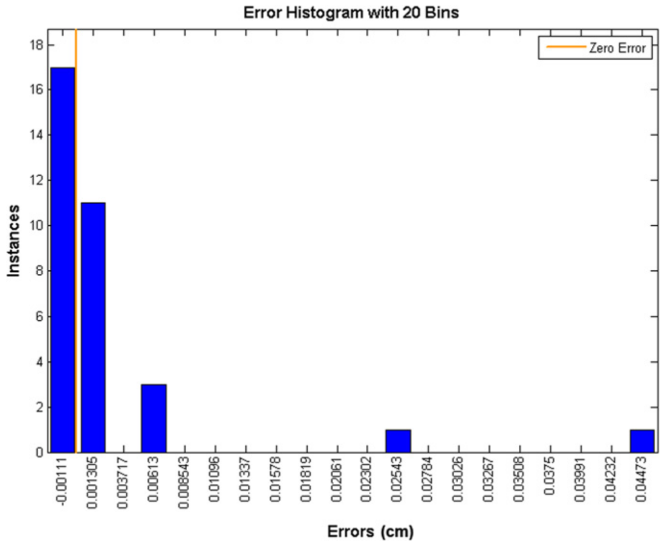

Figure 8. Error histogram plot for the best ANN with 30 neurons in the hidden layer after excessive retraining.

when the network was further retrained we achieved a MSE value of 8.97E-05 cm^2 and a regression value of 0.9992 (ANN number 13 in table 3), which is also shown in figures 7–9. This is probably due to over-fitting and/or larger training subset, which requires more test data to be used for additional checks.

Table 5 shows the error values obtained from the best network result with the 10 neurons in the hidden layer. The maximum errors were −0.17 cm in the LR direction, 0.26 cm in the SI direction and −0.07 cm in the AP direction. In 70% of the cases, the maximum error was in the craniocaudal direction, mostly because tumor motion values in that direction had a widest span of values for the examined cases.

4. Discussion

Our preliminary work (Jurkovic *et al* 2016) showed that there is a preferred tumor motion direction (left, inferior, regardless of the location), more specifically, upper/middle tumors move predominantly left (67%), anterior (67%) and inferior (83%), while lower lesions tend to move more left (60%), posterior (60%) and inferior (60%). Overall, for all the cases, displacement was predominantly left (64%), anterior (55%)

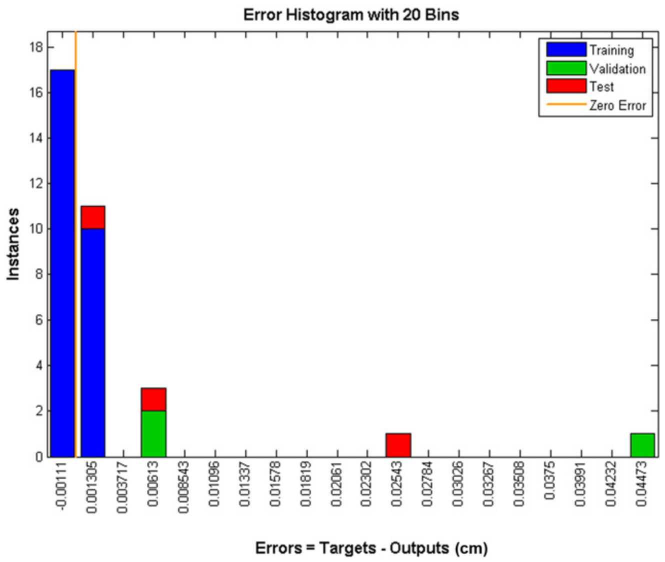

Figure 9. Error histogram plot for the best ANN with 30 neurons in the hidden layer with the partitions also shown.

Table 5. Motion extent error values for the best ANN in cm.

Direction/ patient #	1	2	3	4	5	6	7	8	9	10	11	Average
LR	0.00	0.00	0.01	−0.17	0.01	0.03	0.13	0.03	0.00	0.00	0.00	0.005
SI	−0.02	0.00	0.00	0.26	0.01	−0.02	−0.07	0.05	−0.01	−0.04	−0.03	0.012
AP	0.00	0.00	0.00	−0.01	−0.03	−0.01	−0.03	−0.07	0.01	−0.01	0.02	−0.012

and inferior (73%). This was confirmed by the largest volume change in those directions (Jurkovic *et al* 2016), where for each patient, tumor volume residuals (FUS (volume that is sum of all tumor volumes through the breathing phases)—AIP (volume delineated on the AIP CT dataset)) were calculated for each of the plane halves. The results show that in the majority of the cases the FUS residual volume part prevailed on the following directions: left, 82%, posterior, 55%, and inferior, 64%. The preferred tumor motion that was observed relates to the motion of the tumor volume COM in the AIP CT dataset against the COM in each phase CT dataset. More specifically, it was found that smaller volume sizes require contouring on all the phases since contouring only on the AIP and MIP scans will not cover the extent of motion and volume change in these cases. In other clinical cases, it was found that there is 3D angle similarity when plane fitting is done that depends on the tumor size and location and this can allow for adequate margin calculation when certain parameters are known without extra contouring on all the phases; for example it was found that the lower located tumors have AP angles around 30° and LR-AP angles around 50° and more specifically in all the studied examples correlation was found between the best line fit and the tumor location. In most instances the R^2 value was greater than 0.9 for those planes. Hysteresis (the difference between the inhalation and exhalation trajectory of the tumor) is studied in various publications (Seppenwoolde *et al* 2002, Mageras *et al* 2004, Low *et al* 2005, Boldea *et al* 2008, White *et al* 2013) and represents an important issue for the patients with lung cancer. In most studies computing hysteresis between the trajectories came to calculating the maximum distance between inhalation polygonal curve and exhalation polygonal curve, which can be done by using different distance measures, i.e. Frechet distance for example. In our study we used similar approach and found that there is correlation between the motion trajectories among individual patients depending on the tumor size and location and also between the inhale and exhale paths in some patients that allows for contouring on either just the inhale or exhale phase (Jurkovic *et al* 2016). This finding may lead to a reduction of the work labor especially for smaller tumors where contouring on all the phases is recommended.

Taking into account these findings, the correlation that is found between various factors was further explored, and used as a basis for the neural network creation and design.

Once trained, ANN performed well with regression values that included all three subsets (training, validation and test), which were above 0.80 in almost all cases regardless of the number of input parameters chosen. Some studies point out that a neural network may be an unstable predictor or classifier as it may have high error in the test datasets due to the over fitting on the training dataset (i.e. small changes in initial conditions can lead to high variability in predictions). To overcome this problem a method for reducing variance is suggested. This method is called bagging, which works best with unstable models but can degrade performance of the stable models (Breiman 1998). Bagging is performed using a bootstrap model where bootstrap samples of a training set using sampling with replacement are created. Each bootstrap sample is used to train a different component of base classifier. However, our network showed good performance even with the test datasets, constantly achieving regression values above 0.80 and MSE values below 0.005 cm^2 when all the input parameters were used for the network design.

In MATLAB's default network setup (MATLAB & Simulink. n.d.) each network is trained starting from different initial weights and biases, and with a different division of the first dataset into training, validation, and test sets. Note that the test sets are a good measure of generalization for each respective network, but not for all the networks, because data that is a test set for one network will likely be used for training or validation by other neural network runs. This is why it is recommended to divide the original dataset into two parts, to ensure that a completely independent test set is preserved. However, in the case of a limited size dataset this is not possible. As a consequence, with each run, data is divided randomly and differently for each setup leading to different results, i.e. MSE and R values. For each of the setups multiple runs are performed and then the one with the best overall performance is chosen, which does not necessarily mean that due to the fact that different divisions among data were applied we do not run into overfitting in some of the runs. In our case, we did not have an independent dataset to further check the network's behavior. We also followed these rules to assess the reasonability of the results: the final MSE is small and the test set error and the validation set error have similar characteristics (figure 3). If the network performance is not satisfactory it is recommended to increase the number of hidden neurons and/or increase the number of input values.

Another problem is overgeneralization (MATLAB & Simulink. n.d.). The purpose of training a feedforward network is to modify weight values iteratively such that the weights, ultimately, converge on a solution that correctly relates inputs to outputs. It is normally desirable during the training of a network to be able to generalize basic relations between inputs and outputs based on training data, which may not consist of all possible inputs and outputs for a given problem. A problem that can arise in the training of a neural network involves the tuning of weight values so closely to the training data that the usefulness of the network in processing new data is diminished, which results in over-generalization (or over-training). Basically, this means that the network training may incorporate features of the training dataset that are uncharacteristic of the data as a whole. However, as the measured error continually decreases, the network usefulness and capabilities will also be decreasing (as the network modifies weights to match the characteristics of the training data). This is also where validation and test sets come into the picture, since the network's MSE and R values are a result of the whole network's behavior, i.e. even though the MSE values that are result of the validation and tests dataset may be lower, the overall MSE value (due to the high training set MSE value) may be high enough for the network run to pass our criteria.

Nevertheless, we need to point out limits of our approach. The data set used was small and by increasing the number of neurons in such limited data set size we run into the issue of over fitting. In order to improve the results of the motion extent prediction designed network, the dataset size should be increased. The inclusion of the larger datasets would also give more accurate and stable results. An issue to be discussed concerns the number of hidden neurons used during each trial. However, multiple published studies (Elisseeff and Paugam-Moisy 1996, Lawrence *et al* 1998, Basheer and Hajmeer 2000, Priddy and Keller 2005, Devaraj *et al* 2007, Kuncheva 2012, Sheela *et al* 2013, Mozer *et al* 2014) show that in most of the cases several rules of thumb are suggested, but the most popular approach for finding the optimal number of hidden nodes is by trial and error with one of the mentioned rules as starting point, i.e. retrain the network with varying numbers of hidden neurons and observe the output error as a function of the number of hidden neurons. Furthermore, a study done by Tetko *et al* (1995) states that real examples suggest a rather wide tolerance of ANN to the number of the hidden layer neurons. From our study it is apparent that even with the limited size of the presented dataset, a solution with acceptable accuracy could be found. Once network is build and trained, the results can be used for the adequate margin design.

The amount of the input data can be as large as needed and/or deemed necessary. We have shown that we can achieve good results with only two input neurons, but this is most probably due to the fact that for a particular run the training dataset was much larger than the validation and testing datasets, and the network may have simply over fit the data, which emphasizes our conclusion of more cases needed to test network's stability and accuracy. Besides the data that was already taken into account, tumor characteristics can be further stratified to include pathology, tumor stage, attachment to rigid structures, simulation setup (compression being used or not), etc.

5. Conclusions

The present study discussed the concepts related to neural networks and proposes the use of a given ANN algorithm together with the clinical lung tumor information for prediction of the tumor motion extent. Based on the analysis of our results the proposed solution has several advantages—automated motion extent prediction using the ANN algorithm, usage of the readily available clinical data, and possible high prediction accuracy. In the future, we aim at incorporating more clinical tumor information with the application of different algorithms on the proposed platform, use a larger data set, and carry out additional studies to further improve their liability and stability of the proposed neural network.

Acknowledgments

Author Ines-Ana Jurkovic, Author Sotirios Stathakis, Author Nikos Papanikolaou, and Author Panayiotis Mavroidis declare that they have no conflict of interest. This research received no specific grant from any funding agency in the public, commercial, or not for profit sectors.

References

Ahmad F, Roy K, O'Connor B, Shelton J, Dozier G and Dworkin I 2014 Fly wing biometrics using modified local binary pattern, SVMs and random forest *Int. J. Mach. Learn.Comput.* **4** 279–85

Basheer I A and Hajmeer M 2000 Artificial neural networks: fundamentals, computing, design, and application *J. Microbiol. Methods* **43** 3–31

Beale M H, Hagan M T and Demuth H B 2010 Neural Network Toolbox 7 User's Guide, MathWorks

Boldea V, Sharp G C, Jiang S B and Sarrut D 2008 4D-CT lung motion estimation with deformable registration: quantification of motion nonlinearity and hysteresis *Med. Phys.* **35** 1008–18

Bottaci L, Drew P J, Hartley J E, Hadfield M B, Farouk R, Lee P W, Macintyre I M, Duthie G S and Monson J R 1997 Artificial neural networks applied to outcome prediction for colorectal cancer patients in separate institutions *Lancet* **350** 469–72

Brandner E D, Wu A, Chen H, Heron D, Kalnicki S, Komanduri K, Gerszten K, Burton S, Ahmed I and Shou Z 2006 Abdominal organ motion measured using 4D CT *Int. J. Radiat. Oncol. Biol. Phys.* **65** 554–60

Breiman L 1998 *Arcing Classifiers Ann. Stat.* **26** 801–24

Burke H B 1994 Artificial neural networks for cancer research: outcome prediction *Semin. Surg. Oncol.* **10** 73–9

Chui C-S, Yorke E and Hong L 2003 The effects of intra-fraction organ motion on the delivery of intensity-modulated field with a multileaf collimator *Med. Phys.* **30** 1736–46

Cox J D, Schechter N R, Lee A K, Forster K, Stevens C W, Ang K-K, Komaki R, Liao Z and Milas L 2003 Uncertainties in physical and biological targeting with radiation therapy *Rays* **28** 211–5

Devaraj D, Roselyn J P and Rani R U 2007 Artificial neural network model for voltage security based contingency ranking *Appl. Soft Comput.* **7** 722–7

Dogra H K, Hasan Z and Dogra A K 2013 Face expression recognition using scaled-conjugate gradient back-propagation algorithm *Int. J. Mod. Eng. Res.* **3** 1919–22

Elisseeff A and Paugam-Moisy H 1996 *Size of Multilayer Networks for Exact Learning: Analytic Approach* (*EcoleNormaleSupérieure de Lyon. Laboratoire de l'Informatique du Parallélisme [LIP]*)

Gavin Henri P 2011 The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems (http://people.duke.edu/~hpgavin/ce281/lm.pdf)

George R, Keall P J, Kini V R, Vedam S S, Siebers J V, Wu Q, Lauterbach M H, Arthur D W and Mohan R 2003 Quantifying the effect of intrafraction motion during breast IMRT planning and dose delivery *Med. Phys.* **30** 552–62

Jurkovic I-A, Papanikolaou N, Stathakis S, Li Y, Patel A, Vincent J and Mavroidis P 2016 Assessment of lung tumour motion and volume size dependencies using various evaluation measures *J. Med. Imaging Radiat. Sci.* in press (doi:10.1016/j.jmir.2015.11.003)

Kourou K, Exarchos T P, Exarchos K P, Karamouzis M V and Fotiadis D I 2015 Machine learning applications in cancer prognosis and prediction *Comput. Struct. Biotechnol. J.* **13** 8–17

Kuncheva L 2000 *Fuzzy Classifier Design* (Berlin: Physica-Verlag)

Lawrence S, Giles C L and Tsoi A C 1998 What Size Neural Network Gives Optimal Generalization? Convergence Properties of Backpropagation Online: http://drum.lib.umd.edu/handle/1903/809

Liu H H *et al* 2007 Assessing respiration-induced tumor motion and internal target volume using four-dimensional computed tomography for radiotherapy of lung cancer *Int. J. Radiat. Oncol. Biol. Phys.* **68** 531–40

Low D A, Parikh P J, Lu W, Dempsey J F, Wahab S H, Hubenschmidt J P, Nystrom M M, Handoko M and Bradley J D 2005 Novel breathing motion model for radiotherapy *Int. J. Radiat. Oncol. Biol. Phys.* **63** 921–9

Lujan A E, Larsen E W, Balter J M and Haken R K T 1999 A method for incorporating organ motion due to breathing into 3D dose calculations *Med. Phys.* **26** 715–20

Mageras G S *et al* 2004 Measurement of lung tumor motion using respiration-correlated CT *Int. J. Radiat. Oncol. Biol. Phys.* **60** 933–41

MATLAB & Simulink. n.d. Improve Neural Network Generalization and Avoid Overfitting (http://mathworks.com/help/nnet/ug/improve-neural-networkgeneralization-and-avoid-overfitting.html)

Møller M F 1993 A scaled conjugate gradient algorithm for fast supervised learning *Neural Netw.* **6** 525–33

Mozer M C, Smolensky P, Touretzky D S, Elman J L and Weigend A S 2014 *Proc. 1993 Connectionist Models Summer School* (Oxford: Psychology Press)

Naqvi S A and D'Souza W D 2005 A stochastic convolution/superposition method with isocenter sampling to evaluate intrafraction motion effects in IMRT *Med. Phys.* **32** 1156–63

Othman I B and Ghorbel F 2014 Stability evaluation of neural and statistical classifiers based on modified semi-bounded plug-in algorithm *Int. J. Neural Netw. Adv. Appl.* **1** 37–42

Priddy K L and Keller P E 2005 *Artificial Neural Networks* (Bellingham, WA: SPIE)

Seppenwoolde Y, Shirato H, Kitamura K, Shimizu S, van Herk M, Lebesque J V and Miyasaka K 2002 Precise and real-time measurement of 3D tumor motion in lung due to breathing and heartbeat, measured during radiotherapy *Int. J. Radiat. Oncol. Biol. Phys.* **53** 822–34

Sheela K G, Deepa S N, Sheela K G and Deepa S N 2013 Review on methods to fix number of hidden neurons in neural networks *Math. Probl. Eng.* **2013** e425740

Shiffman D 2012 *The Nature of Code: Simulating Natural Systems with Processing* (The Nature of Code)

Shirato H, Seppenwoolde Y, Kitamura K, Onimura R and Shimizu S 2004 Intrafractional tumor motion: lung and liver *Semin. Radiat. Oncol.* **14** 10–8

Shirato H *et al* 2006 Speed and amplitude of lung tumor motion precisely detected in four-dimensional setup and in real-time tumor-tracking radiotherapy *Int. J. Radiat. Oncol. Biol. Phys.* **64** 1229–36

Sonke J-J, Lebesque J and van Herk M 2008 Variability of four-dimensional computed tomography patient models *Int. J. Radiat. Oncol. Biol. Phys.* **70** 590–8

Tetko I V, Livingstone D J and Luik A I 1995 Neural network studies: I. Comparison of overfitting and overtraining *J. Chem. Inf. Comput. Sci.* **35** 826–33

Underberg R W M, Lagerwaard F J, Cuijpers J P, Slotman B J, van Sörnsen de Koste J R and Senan S 2004 Four-dimensional CT scans for treatment planning in stereotactic radiotherapy for stage I lung cancer *Int. J. Radiat. Oncol. Biol. Phys.* **60** 1283–90

Wei J S *et al* 2004 Prediction of clinical outcome using gene expression profiling and artificial neural networks for patients with neuroblastoma *Cancer Res.* **64** 6883–91

White B, Zhao T, Lamb J, Wuenschel S, Bradley J, El Naqa I and Low D 2013 Distribution of lung tissue hysteresis during free breathing *Med. Phys.* **40** 043501