Development of R-Shiny interface for implementation of backpropagation neural network model in breast cancer classification

G Islahuzaman, R Santoso, B Warsito, D Ispriyanti, H Yasin

Statistics Department, Faculty of Science and Mathematics, Diponegoro University, Semarang Indonesia

Corresponding author: gustiyan98@gmail.com

Abstract. Artificial Neural Network or Neural Network (NN) is an information processing system that has similar characteristics to the neural network in living things. One type of NN that is often used in classification is Backpropagation Neural Network (BPNN). BPNN is an NN model that is often used for classification because it does not need to use assumptions and has high accuracy. One of the classification problems that can be solved with BPNN is the classification of breast cancer. The breast cancer data used in this study came from the UCI Machine Learning website. The problem with BPNN is that programming is difficult for users who do not understand the program, especially the R program. Therefore, to make it easier for users to analyze BPNN, an R-Shiny application or interface is created using the RStudio program. The application or R-Shiny interface that has been created has several advantages, namely the application process that is fast in displaying classification results, the use of user-friendly applications and the use of applications that are more comfortable when compared to having to write syntax such as in the R program. BPNN classification results use The R-Shiny interface has a different level of accuracy for each experiment due to the random distribution of training & testing data. The experiments conducted in this study resulted in a range of accuracy values ranging from 58.33% to 91.67% with an average accuracy of 74.17%.

1. Introduction
Cancer is one of the deadliest diseases in the world. Every year there are millions of people who die from cancer. One type of cancer that often attacks humans is breast cancer. Breast cancer is a type of malignant tumor that develops in breast cells [2]. Breast cancer is still a major health problem for women around the world. The incidence of this disease has continued to increase in the last 10 years in various parts of the world. Breast cancer can be detected by doing mammography. However, it requires a relatively high cost to do this. Therefore, it is necessary to use other alternatives that are accurate and easy in detecting breast cancer without expensive costs. One method that can be used to solve these problems is to use an artificial neural network.

Artificial Neural Network (ANN) or Neural Network (NN) is an information processing system that has similar characteristics to the neural network in living things. This study uses a neural network model, namely the Backpropagation Neural Network (BPNN) model to classify data. BPNN is an NN model that is often used for classification because it does not need to use assumptions and has high accuracy. BPNN is a type of supervised learning, which means that the output of the network is...
compared with the expected target so that an output error is obtained, then this error is propagated back to improve the network weight in order to minimize errors [4].

The problem with BPNN is the use of programs that are difficult for users who do not understand the program, especially the R program. One of the programs in R that can create a user interface menu is R-Shiny. R-Shiny is a toolkit of R programs that can be used to create GUI. This research will make an interface with R-Shiny to classify breast cancer. This is intended to make it easier for someone who does not understand programming to make it easier to conduct BPNN analysis, especially in cases of breast cancer.

2. Literature review

2.1. Breast cancer
Breast cancer is still a major health problem for women around the world. The incidence of this disease has continued to increase in the last 10 years in various parts of the world [3]. Based on the latest data from the International Agency for Research on Cancer (IARC) GLOBOCAN in 2018, breast cancer is the fifth leading cause of death worldwide with 627,000 people or 6.6%. Breast cancer is a group of abnormal cells in the breast that continue to multiply. Eventually these cells form a lump in the breast. If the cancer lumps are not removed or uncontrolled, cancer cells can spread to other parts of the body and later can cause death [8].

2.2. Artificial neural network
Artificial Neural Networks (ANN) or commonly called Artificial Neural Networks (ANN) are information processing systems that have similar characteristics to neural networks in living things [1]. In subsequent developments the letter "A" in the term ANN was often omitted to shorten the writing so that it was more popular with the term Neural Network (NN) only. NN was first introduced by Mc. Culloch and Pitts in 1943 is a simple model of a real nerve in the human brain like a binary threshold unit. The principles of Neural Network modelling are developed from the characteristics and workings of the human brain. In processing information, the human brain consists of a number of neurons which perform simple tasks. Due to the interconnectedness of neurons, the human brain can perform very complex processing functions. Information processing can only be done after going through the previous learning process. Information processing in human neural networks is adaptive, which means that the relationship between neurons occurs dynamically, the strength of the relationship between neurons can change from time to time, and has the ability to learn new information [9].

2.3. Backpropagation neural network
Backpropagation is one of the learning algorithms in artificial neural networks. The term backpropagation relates to the method for calculating the gradient of the error function in terms of weights for a feedforward network. Training using backpropagation includes three stages, namely the feed forward of the input pattern, calculation of errors, and adjustment of weights [9].

2.4. Activation function
The activation function in ANN acts as a signal to determine the output to several other neurons. The activation function used in this study is the binary sigmoid function (logistic sigmoid). This is in accordance with the binary classification output of model. The binary sigmoid function (logistic sigmoid) is a function that is commonly used and has a value between 0 and 1 and is used for networks with an output value between 0 and 1. The equation of the binary sigmoid function is defined as

\[ y = \frac{1}{1 + e^{-\alpha x}}, -\infty < x < \infty \]  
(1)
2.5. Neural network model
The network architecture consists of one input layer unit with the number of neurons (nodes) \( p \), one hidden layer with \( n \) units, and one output layer unit consisting of one neuron (node) can be written as a neural network model as follows

\[
y = \psi_o \left( w_{co} + \sum_n w_{no} \psi_n \left( w_{cn} + \sum_i w_{in} x_i \right) \right)
\]

where \( y \) is the output value, \( \psi_o \) is the output activation function, \( w_{co} \) is the constant weight from hidden layer to output layer, \( w_{no} \) is the neuron weight from \( n^{th} \) node to output, \( \psi_n \) is the activation function in the hidden layer, \( w_{cn} \) is the constant weight in the hidden layer, \( w_{in} \) is the neuron weight from \( i^{th} \) input to \( n^{th} \) node, and \( x_i \) is \( i^{th} \) input.

2.6. Backpropagation training algorithm
The BPNN training algorithm consists of three stages, namely phase I forward propagation, phase II back propagation and phase III weight change. The algorithm of the backpropagation neural network according to Fausett (1994) is as follows [10].

1. Give the initial value of the weight with a small random number, namely between \(-1\) to \(1\).
2. Setting a stop condition by providing an error limit value or the maximum number of epochs (iterations). As long as the stopping condition has not been achieved, then carry out steps 2 to 9.
3. For each training process, steps 3 to 8 are carried out.

Phase I: Forward Propagation (Feed Forward)
4. Each input unit \( x_i, i = 1, 2, \ldots, n \) sends an input signal to each input in the hidden layer.
5. Each hidden layer unit \( z_j, j = 1, 2, \ldots, p \) is multiplied by its weight \( w_{ji} \) and added to its bias weight \( w_{j0} \).

\[
z_{inj} = w_{j0} + \sum_{i=1}^{n} x_i w_{ji}
\]

Then calculate the output signal from the hidden layer using the activation function.

\[
z_j = f(z_{inj}) = \frac{1}{1 + e^{-z_{inj}}}
\]

The output signal is then forwarded to each output unit.
6. Each unit of output \( y_k, k = 1, 2, \ldots, m \) is multiplied by the weight \( v_{kj} \) and added to the bias \( v_{k0} \).

\[
y_{netk} = v_{k0} + \sum_{j=1}^{p} \sum_{k=1}^{m} z_j v_{kj}
\]

Use the activation function to calculate the output signal.

\[
y_k = f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}}
\]

Phase II: Backpropagation (error from backpropagation)
7. Each output unit \( y_k, k = 1, 2, \ldots, m \) receives a target pattern \( t_k \) according to the input pattern during training and then the output layer error information \( \delta_k \) is calculated, \( \delta_k \) is sent to the layer below it and is used to calculate the amount of weight and bias correction (\( \Delta v_{kj} \) dan \( \Delta v_{k0} \)) between the hidden layer and the output layer.
\[ \delta_k = (t_k - y_k) f'(y_{net_k}) \] (7)
\[ \delta_k = (t_k - y_k) y_k (1 - y_k) \] (8)

Calculate the weight change rate \((v_{kj})\) and the change in bias \((v_{ko})\) with learning rate \((\alpha)\).
\[ \Delta v_{kj} = \alpha \delta_k z_{j}; k = 1, 2, \ldots, m; j = 0, 1, \ldots, p \] (9)
\[ \Delta v_{ko} = \alpha \delta_k \] (10)

8. For each unit in the hidden layer (from 1-unit to p-unit; \(i: 1, 2, \ldots, n; k: 1, 2, \ldots, m\)) the hidden layer error information \((\delta_j)\) is calculated. The error information is used to calculate the amount of weight and bias correction \((\Delta w_{ji} \text{ and } \Delta w_{jo})\) between the input layer and the hidden layer.
\[ \delta_{net_j} = \sum_{k=1}^{m} \delta_k v_{kj} \] (11)

Calculate the \(\delta_j\) factor in the hidden unit using the derived activation function.
\[ \delta_j = \delta_{net_j} f'(z_{in_j}) \] (12)
\[ \delta_j = \delta_{net_j} z_{j} (1 - z_{j}) \] (13)

Calculate the rate of change in the weight of \(w_{ji}\) and the change in bias \(w_{jo}\) with the learning rate \((\alpha)\).
\[ \Delta w_{ji} = \alpha \delta_j x_i \] (14)
\[ \Delta w_{jo} = \alpha \delta_j \] (15)

Phase III: Changes in weight and bias

9. Each unit of output \(y_k (k: 1, 2, \ldots, m)\) changes its bias and weight \((j: 0, 1, \ldots, p)\) to produce new weights and biases.
\[ v_{kj}(\text{new}) = v_{kj}(\text{old}) + \Delta v_{kj} \] (16)

Likewise, for each hidden unit starting from the 1st unit to the p-unit the weight and bias are changed.
\[ w_{ji}(\text{new}) = w_{ji}(\text{old}) + \Delta w_{ji}; \text{ } j = 1, 2, \ldots, p \text{ and } i = 1, 2, \ldots, n \] (17)

10. The training process or training will stop when the stop conditions have been met, namely when it has reached the maximum epoch (iteration).

2.7. Classification

Classification is a process of finding a set of models that explain and differentiate data classes, so that the model can be used to predict the value of an unknown class on an object [9]. One of the methods used to record the classification results is the confusion matrix. The confusion matrix is a table that records the results of classification work [6]. The confusion matrix used in disease classification has 6 measures, namely sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), prevalence, and accuracy [7].
### Table 1. Confusion Matrix

| Test    | Positive | Negative | Total |
|---------|----------|----------|-------|
| Positive| A        | B        | A+B   |
| Negative| C        | D        | C+D   |
| Total   | A+C      | B+D      | A+B+C+D |

1. Sensitivity = \( \frac{A}{A+C} \)
2. NPV = \( \frac{D}{C+D} \)
3. PPV = \( \frac{A}{A+B} \)
4. Prevalence = \( \frac{A+C}{A+B+C+D} \)
5. Specificity = \( \frac{D}{B+D} \)
6. Accuracy = \( \frac{A+D}{A+B+C+D} \)

2.8. **R-Shiny**

R-Shiny is one of the toolkits of the R program using the shiny package which is used to create interfaces such as GUIs and interactive webs. During its development, the shiny package is more often used in the RStudio program, which is one of the development programs of R. RStudio itself has advantages compared to ordinary R, namely having a user-friendly appearance and toolkit and easier to use by general users. R-Shiny consists of two main components, namely ui.R and server.R.

### 3. Research methods

#### 3.1. Types and sources of data

The type of data used in this study is secondary data taken from the UCI Machine Learning website (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra). The data is the 2018 Coimbra Breast Cancer dataset which came from Miguel Patrício, José Pereira, Joana Crisóstomo, Paulo Matafome, Raquel Seiça, Francisco Caramelo from the Faculty of Medicine of the University of Coimbra and Manuel Gomes from the University Hospital Center of Coimbra. The data were collected from 116 patients (64 cancer positive patients and 52 cancer negative patients), and there were 10 variables. These variables are as follows:

1. Cancer status as a response variable (Y)
2. Age as the first independent variable (X₁)
3. BMI as the second independent variable (X₂)
4. Glucose as the third independent variable (X₃)
5. Insulin as the fourth independent variable (X₄)
6. HOMA as the fifth independent variable (X₅)
7. Leptin as the sixth independent variable (X₆)
8. Adiponectin as the seventh independent variable (X₇)
9. Resistin as the eighth independent variable (X₈)
10. MCP.1 as the ninth independent variable (X₉)

#### 3.2. Data analysis method

The software used in the following research is RStudio version 1.2.5019 using the neuralnet package to perform the neural network calculation process and the shiny package to create a GUI. The steps taken in this research are as follows:
1. Describe the data,
2. Creating a layout design for the R-Shiny interface, namely ui.R,
3. Creating a server.R program on the R-Shiny interface,
4. Run the application or R-Shiny interface that has been created,
5. Determine the response variable, determine the percentage of training data, the value of learning rate, error tolerance, maximum iteration, the activation function used, the number of neurons in the hidden layer and the maximum number of iterations.
6. Creating a BPNN model using training data,
7. Repeat from step 5 to step 6 if iteration has exceeded the maximum number of iterations. If the number of iterations less or equal to the maximum number of iterations then continue to the next step.
8. Testing data testing based on the training data model.
   a. Calculating the accuracy value
   b. Calculate the specificity value
   c. Calculating the prevalence value
   d. Calculating the PPV value
   e. Calculating the sensitivity value
   f. Calculating the NPV value
9. Comparing the accuracy results to obtain the best model.

**4. Results and discussion**

**4.1. Architectural design of BPNN**

The backpropagation neural network or BPNN has a network architecture consisting of 3 layers, namely the input layer, hidden layer, and output layer. The network architecture designed in this study consists of 3 layers (input, hidden, and output) with 9 neurons for the input layer, for the hidden layer there are 4 choices for the number of neurons, namely 5, 10, 15 and 20 neurons, and for the output layer numbering 1 neuron. The following is an example of an architectural design that will be formed with 5 neurons hidden layers.

![BPNN Architectural Design](image)

**Figure 1 BPNN Architectural design**

**4.2. R-Shiny interface design**

The R-Shiny interface for the implementation of the Backpropagation Neural Network model in the classification of breast cancer was built using the Shiny package in the RStudio version 1.2.5019 application. There are two important parts in creating an interface or GUI, the first part is designing the layout or appearance of the GUI (ui.R) and the second part is designing the GUI program (server.R). The R-Shiny interface consists of six main menus, namely the data menu, the analysis menu, the results menu, the predictive value table menu, the plot menu, and the breast cancer prediction menu and there is a side panel used for data input.
4.3. Illustrations using the R-Shiny interface
This R-Shiny interface is used to classify breast cancer using the BPNN method where users only need to press a button and type a numeric input value. In the process of classifying breast cancer data using an application or interface that has been made, it only requires a few simple steps to classify. The Data menu is used to display the data that has been entered in the sidebar panel. The Analysis menu is used to enter parameters which include selecting response variables, activation functions, determining the percentage of training data, the number of neurons in the hidden layer, the maximum number of iterations, the error limit value, the learning rate value and the random seed value. The Results menu is used to determine the classification results and weights obtained from the BPNN method. Prediction Value Table menu is prepared to determine the predictive value of breast cancer patient status, which consists of 2 data, namely training data and testing data. In the Plot menu, you can see the Backpropagation neural network architecture image, and the Breast Cancer Prediction menu is used to predict new data using the previously trained BPNN model.

4.4. Classification results of breast cancer
BPNN learning outcomes form the R-Shiny application or interface with a variety of different network architecture variations so that the best model can be obtained. The best model can be used as a basis for a more accurate classification of a person's breast cancer. The following are the results of BPNN
learning with various network architecture variations using 4 percentages of training and testing data, 4 numbers of neurons in the hidden layer and 2 learning rate values ($\alpha$).

| Percentage of Data (%) | Number of Neurons in Hidden Layer | Learning Rate Value ($\alpha$) | Accuracy of Test Data | 1 | 2 | 3 | 4 | 5 | Average |
|------------------------|-----------------------------------|-------------------------------|----------------------|---|---|---|---|---|---------|
| 60:40                  | 5                                 | 0.1                           | 68.09%               | 74.47% | 72.34% | 76.60% | 68.09% | 71.92% |
| 60:40                  | 5                                 | 0.5                           | 59.57%               | 57.45% | 70.21% | 61.70% | 74.47% | 64.68% |
| 60:40                  | 10                                | 0.1                           | 74.47%               | 68.09% | 65.96% | 72.34% | 80.85% | 72.34% |
| 60:40                  | 10                                | 0.5                           | 59.57%               | 65.96% | 76.60% | 59.57% | 68.09% | 68.94% |
| 60:40                  | 15                                | 0.1                           | 63.83%               | 72.34% | 74.47% | 68.09% | 74.47% | 70.64% |
| 60:40                  | 15                                | 0.5                           | 78.72%               | 70.21% | 78.72% | 65.96% | 78.72% | 74.47% |
| 60:40                  | 20                                | 0.1                           | 65.96%               | 76.60% | 68.09% | 70.21% | 72.34% | 70.64% |
| 60:40                  | 20                                | 0.5                           | 72.34%               | 70.21% | 74.47% | 85.11% | 68.09% | 74.04% |
| 70:30                  | 5                                 | 0.1                           | 71.43%               | 82.86% | 74.29% | 71.43% | 54.29% | 70.86% |
| 70:30                  | 5                                 | 0.5                           | 59.57%               | 74.29% | 65.71% | 74.29% | 54.29% | 70.86% |
| 70:30                  | 10                                | 0.1                           | 68.57%               | 82.86% | 77.14% | 74.29% | 77.14% | 76.00% |
| 70:30                  | 10                                | 0.5                           | 71.43%               | 80.00% | 62.86% | 82.86% | 82.86% | 76.00% |
| 70:30                  | 15                                | 0.1                           | 68.57%               | 82.86% | 71.43% | 74.29% | 77.14% | 74.86% |
| 70:30                  | 15                                | 0.5                           | 77.14%               | 77.14% | 80.00% | 77.14% | 65.71% | 75.43% |
| 70:30                  | 20                                | 0.1                           | 60.00%               | 77.14% | 77.14% | 71.43% | 74.29% | 72.00% |
| 70:30                  | 20                                | 0.5                           | 65.71%               | 71.43% | 57.14% | 65.71% | 68.57% | 65.71% |
| 80:20                  | 5                                 | 0.1                           | 54.17%               | 58.33% | 66.67% | 50.00% | 70.83% | 60.00% |
| 80:20                  | 5                                 | 0.5                           | 66.67%               | 75.00% | 75.00% | 66.67% | 79.17% | 72.50% |
| 80:20                  | 10                                | 0.1                           | 75.00%               | 79.17% | 70.83% | 79.17% | 45.83% | 70.00% |
| 80:20                  | 10                                | 0.5                           | 75.00%               | 66.67% | 87.50% | 62.50% | 75.00% | 73.33% |
| 80:20                  | 15                                | 0.1                           | 79.17%               | 70.83% | 79.17% | 62.50% | 75.00% | 73.33% |
| 80:20                  | 15                                | 0.5                           | 79.17%               | 75.00% | 83.33% | 79.17% | 75.00% | 70.00% |
| 80:20                  | 20                                | 0.1                           | 83.33%               | 75.00% | 66.67% | 83.33% | 75.00% | 76.67% |
| 80:20                  | 20                                | 0.5                           | 79.17%               | 79.17% | 70.83% | 62.50% | 70.83% | 72.00% |
| 90:10                  | 5                                 | 0.1                           | 66.67%               | 58.33% | 58.33% | 66.67% | 66.67% | 63.33% |
| 90:10                  | 5                                 | 0.5                           | 83.33%               | 83.33% | 83.33% | 66.67% | 66.67% | 76.67% |
| 90:10                  | 10                                | 0.1                           | 66.67%               | 83.33% | 75.00% | 66.67% | 66.67% | 71.67% |
| 90:10                  | 10                                | 0.5                           | 83.33%               | 75.00% | 50.00% | 91.67% | 50.00% | 91.67% | 78.33% |
| 90:10                  | 15                                | 0.1                           | 58.33%               | 83.33% | 91.67% | 83.33% | 66.67% | 76.67% |
| 90:10                  | 15                                | 0.5                           | 75.00%               | 75.00% | 83.33% | 50.00% | 71.67% |
| 90:10                  | 20                                | 0.1                           | 83.33%               | 58.33% | 83.33% | 83.33% | 58.33% | 73.33% |
| 90:10                  | 20                                | 0.5                           | 41.67%               | 75.00% | 75.00% | 91.67% | 66.67% | 70.00% |

Table 2 shows that the most optimal network architecture in the classification of breast cancer is the percentage of training and testing data is 90:10 (104 training data and 12 testing data), the number of neurons in the hidden layer is 10 neurons and the learning rate ($\alpha$) is equal to 0.5 because it has the highest average accuracy of 78.33%. The optimal network architecture will be used to classify breast cancer. The following are the results of the classification of breast cancer on the R-Shiny interface with 10 trials using the most optimal network architecture.
Table 3. Classification results

| Experiment | Level of Test Data Accuracy |
|------------|-----------------------------|
| 1          | 66.67%                      |
| 2          | 75.00%                      |
| 3          | 75.00%                      |
| 4          | 66.67%                      |
| 5          | 91.67%                      |
| 6          | 75.00%                      |
| 7          | 75.00%                      |
| 8          | 66.67%                      |
| 9          | 91.67%                      |
| 10         | 58.33%                      |
| **Average** | **74.17%**                  |

Table 3 shows the classification results on the R-Shiny application or interface that has been made to have different levels of accuracy in the 10 experiments. The results of the classification of breast cancer have the smallest test data accuracy rate of 58.33% and the highest test data accuracy rate of 91.67% with an average of 74.17%.

5. Conclusion

From the results of research and discussion on the development of the R-Shiny application or interface used for the implementation of the backpropagation neural network model in the case example of breast cancer classification, it can be concluded that the R-Shiny application or interface that has been made has several advantages, including a fast application process. In displaying the results of breast cancer classification, the use of a user-friendly application that is easy for users to understand and the use of applications that is more comfortable when compared to having to write syntax as in program R. The classification results of BPNN using the R-Shiny interface have different levels of accuracy in each experiment. Due to the random distribution of training & testing data. The experiments conducted in this study resulted in a range of accuracy values ranging from 58.33% to 91.67% with an average accuracy of 74.17%.

References

[1] Fausett L 1994 Fundamentals of Neural Networks: Architectures, Algorithms, and Applications (New Jersey: Prentice-Hall Inc.)
[2] Halodoc 2019 Kanker Payudara https://www.halodoc.com/kesehatan/kanker-payudara. (Accessed : 19 March 2020)
[3] Ika 2016 Jumlah Pasien Terdiagnosis Kanker Payudara Terus Meningkat. https://ugm.ac.id/id/berita/12473-jumlah-pasien-terdiagnosis-kanker-payudara-terus- meningkat (Accessed : 19 March 2020)
[4] Jumarwanto A, Hartanto R and Prastiyanto D 2009 Jurnal Teknik Elektro 1 11-21
[5] Patrício M, Pereira J, Crisóstomo J, Matafome P, Gomes M, Seiça R and Caramelo F 2018 BMC Cancer 18 29
[6] Prasetyo E 2012 Data Mining: Konsep Dan Aplikasi Menggunakan Matlab (Yogyakarta: Andi Publisher)
[7] Rasjidi I 2010 Epidemiologi Kanker Pada Wanita (Jakarta: Sagung Seto)
[8] Tapan E 2005 Kanker, Antioksidan, dan Terapi Komplementer (Jakarta: Elex Media Komputindo)
[9] Warsito B 2009 Kapita Selektiva Statistika Neural Network (Semarang: BP UNDIP Semarang)
[10] Zamani A M, Amalah B and Munif A 2012 Jurnal Teknik ITS 1 222-227