Classification of liver cancer with microRNA data using the deep neural network (DNN) method

O H Purba, E A Sarwoko, Khadijah, Suhartono, and A Wibowo

Department of Informatics/Computer Science, Faculty of Science and Mathematics, Diponegoro University, Jl. Prof. Soedarto, SH, Tembalang, Semarang 50275, Indonesia

Corresponding author: oliverpurba@students.undip.ac.id

Abstract. Liver cancer is one of the highest causes of death in the world. According to IARC data in 2018, the number of cases of liver cancer reached 841,080 cases and 781,631 of them declared dead. One effort that can be done to reduce cases of death from liver cancer is early detection by doing a classification on MicroRNA data. MicroRNA can be used to identify whether a cell is a cancer cell or not at the earliest stages. MicroRNA data studied was obtained from the GDC Data Portal of the National Cancer Institute (NCI). The Deep Neural Network (DNN) method is one of the methods that can be used in classifying cancer. Data normalization in the DNN method brings input values with an unlimited range into a limited range of output values. Data normalization at data preprocessing work more optimally by adding activation functions and Batch Normalization in each hidden layer. Different types of normalization and use of different activation functions can be compared to get the best accuracy value for DNN. This study compares three types of data normalization, namely Min-Max, Sigmoid, and Softmax. Whereas the activation functions being compared are ReLU, Sigmoid, and TanH. The results showed the best accuracy for the classification of liver cancer with MicroRNA data obtained using Min-Max data normalization, ReLU activation function, batch normalization with parameters of 2 hidden layers, 200 epochs, and 0.004 learning rate with an accuracy value reaching 98.33%.

1. Introduction
Liver cancer (Hepatocellular Carcinoma / HCC) is cancer that occurs in liver cells (liver). The cancerous liver will enlarge like a ball in the right upper abdomen, below the diaphragm and above the stomach [1]. According to data from the International Agency for Research on Cancer, liver cancer is the leading cause of death in the world with 841,080 cases and 781,631 of them declared dead [2].

One effort that can be done to reduce the number of cases of cancer deaths is early detection [3]. One way that can be done to do early detection of liver cancer is to classify the MicroRNA data. MicroRNAs (miRNAs) are small, non-coded groups of Ribonucleic Acid (RNA) (ncRNAs) that regulate gene expression by targeting the corresponding carrier RNA (mRNA) [4]. MicroRNA can be used to identify whether a cell is a cancer cell or not even in the earliest stages [5]. Classification is a method used by machines to sort or classify objects based on certain characteristics [6].

MicroRNA classification process can be done in various ways. One classification technique that can be applied is to use the Deep Neural Network (DNN) method. DNN is one of the newest and best machine learning techniques for pattern recognition [7]. DNN has several hidden layers and makes this model an expressive model that can learn the complicated relationship between inputs and outputs.
[8]. This hidden layer will gradually map unorganized low-level features into high-level data representations [7].

Research by (Thomas & Sael, 2017) predicts the sequence of miRNA precursors using the miRBase 18.0 release dataset with a sample data distribution of 60% training, 20% validation and 20% testing, with 58 features indicating that the DNN method has an accuracy of 96.8% [9]. Research by (Joshi & Metha, 2018) shows the DNN method with Linear Discriminant Analysis (LDA) for breast cancer classification gives an accuracy of 97.06% [10].

Research by (Wibowo & Ningrum, 2019) using the DNN method with Batch normalization for breast cancer classification gives an accuracy of 93.67% [11]. Research by (Nurul, et.al, 2012) shows that Min-Max Normalization has the highest accuracy with an average accuracy of 96.86%, in the case of breast cancer classification by BPGDA Neural Network method [12]. Research by (Pedamonti, 2018) using the DNN method with MNIST data shows the ReLU activation function has the lowest error value with a value of 0.0857 [13].

To determine the best level of accuracy, in this research three types of data normalization will be compared, namely Min-Max, Sigmoid, and Softmax, while the activation functions compared are ReLU, Sigmoid, and TanH.

2. Methodology

This research is divided into several stages, namely data collection, data preprocessing, data partition, training, testing, and evaluation. This research stages can be seen in Figure 1 below.

![Figure 1 Research stages](image)

2.1. Data collection

The dataset used in this study represents the value of microRNA expression in liver cancer. The microRNA dataset was obtained from the GDC Data Portal of the National Cancer Institute (NCI, 2018). The data used in this study consisted of 600 data divided into 300 data on liver cancer class and 300 data on normal class with a total of 328 MicroRNA features.

2.2. Data pre-processing

The pre-processing stage will prepare the data so that obtain maximum processing results. In this research, pre-processing consists of two processes namely data mapping and data normalization. Data mapping aims to adjust the data format from the data source to the database. Then data normalization is performed where the values of the data attributes are scaled into a certain range such as 0.0-1.0 or -
1.0-1.0. The data normalization process used in this research are Min-Max, Sigmoid, and Softmax normalization.

2.2.1. Min-max normalization. Min-Max normalization function is one of the methods in data transformation that uses the smallest value data and the largest value data as a range data reference. This method maps the value of \( x \) on \( P \) dataset to \( x' \) in the range of \([\min(P), \max(P)]\) [14]. The min-max normalization formula can be seen in equation 1.

\[
    x' = \frac{(x - \min(P))}{(\max(P) - \min(P))}
\]  

(1)

2.2.2. Sigmoid normalization. Sigmoid normalization function transforms values into a range between 0 and 1 by utilizing the epsilon (\( e \)) value [15]. This function is shown by equation 2.

\[
    \sigma(x) = \frac{1}{1 + e^{-ax}}
\]  

(2)

2.2.3. Softmax normalization. Softmax normalization function transforms the range of values from input \( x \) into a range between 0 and 1 using the sigmoid function by utilizing the mean and standard deviation [15]. This function is shown by equation 3.

\[
    \varphi(x) = \frac{1}{1 + e^{-\frac{x}{\sigma}}}
\]  

(3)

2.3. Data partition

The data partition stages divide the data into training data and test data. Training data is data used for the training process, while test data is data used for the testing process. MicroRNA data will be divided by a ratio of 70% for training data and 30% for test data.

2.4. DNN training

The training algorithm for the DNN method with two hidden layers that applies vectorization are as follows [15]:

DNN method with two hidden layers that applies vectorization is as follows:

1. Step 0: Initialize the weights with small random numbers, specify epoch = 1, the learning rate (\( \alpha \)), and the number of hidden layers.
2. Step 1: If the termination condition is not met, i.e. epoch \( \neq \) maximum epoch, then go to step 2.
3. Forward Propagation Phase
4. Step 2: Each input unit receives a signal and passes it to the hidden unit.
5. Step 3: Each output value at the layer is denoted as \( A^{[l]} \), with \( l = 0, 1, \ldots, p \) and output value unit \( i \) on layer \( l \) is denoted as \( a_i^{[l]} \), with \( i = 1, 2, \ldots, j \). Calculate the output value on the first hidden layer until the last hidden layer with the following steps [16]:
   a. Calculate the value of input weights for \( m \) training data using equation 4.

\[
    Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}
\]  

(4)

b. Calculate the output value on the hidden layer with the ReLU activation function using equation 5, Sigmoid activation function using equation 6, or TanH activation function using equation 7.

\[
    A^{[l]} = f_{ReLU}(Z^{[l]}) = \max(0, Z^{[l]})
\]  

(5)

\[
    A^{[l]} = f_{Sigmoid}(Z^{[l]}) = \frac{1}{1+e^{-Z^{[l]}}}
\]  

(6)

\[
    A^{[l]} = f_{Tanh}(Z^{[l]}) = \frac{(1-e^{-Z^{[l]}})}{(1+e^{-Z^{[l]}})}
\]  

(7)
5. Step 4: Calculate the output value at the output layer with the following steps [16]:
   a. Calculate the weight value on the output layer using equation 8.
   $$Z[p] = W[p], A[p−1] + b[p]$$ (8)
   b. Calculate the output value at the output layer with the Sigmoid activation function using equation 9.
   $$A[p] = \sigma(Z[p]) = \frac{1}{1 + e^{Z[p]}}$$ (9)

6. Step 5: Calculate the error value at the output layer with binary cross entropy using equation 10.
   $$L(A[p], Y) = -\frac{1}{m}\sum_{i=1}^{m}(y_i \log a_i) + ((1 - y_i) \ast (\log(1 - a_i)))$$ (10)

Backward Propagation Phase

7. Step 6: Calculate the value of the change in parameter weights and at the output layer by deriving the function in the forward propagation phase with the following steps [16]:
   a. Calculate the gradient of the loss value at the output layer using equation 11.
   $$dA[p] = \frac{dL(a,y)}{dA[p]}$$ (11)
   b. Calculate the value of $$dZ[p]$$ on the output layer using equation 12.
   $$dZ[p] = dA[p] \ast \sigma’(A[p])$$ (12)
   c. Calculate the value of the parameter derivative $$W$$ using equation 13.
   $$dW[p] = \frac{1}{m} \ast dZ[p] \ast A[p−1]^T$$ (13)
   d. Calculate the value of the parameter derivative $$b$$ using equation 14.
   $$db[p] = \frac{1}{m} \ast \sum_{i=1}^{m} dZ[p]$$ (14)
   e. Calculate the output value at the output layer using equation 15.
   $$dA[p−1] = W[p]^T \ast dZ[p]$$ (15)

8. Step 7: After getting the values from the parameter derivative $$W$$ and $$b$$, calculate the change in parameter weights using the SGD method using equation 16 and equation 17.
   $$W[r] = W[p] - \alpha(dW[p])$$ (16)
   $$b[r] = b[p] - \alpha(db[p])$$ (17)

9. Step 8: Calculate the rate of change of weights in the hidden layer first by reducing the propagation phase advanced functions with the following steps:
   a. Calculate the value of $$dZ[p−1]$$ on the hidden layer using equation 18.
   $$dZ[p−1] = dA[p−1] \ast f'_{ReLU/Sigmoid/Tanh}(A[p−1])$$ (18)
   b. Calculate the value from parameter derivative $$W$$ and $$b$$ using equation 13 and 14.
   c. Calculate the output value in the hidden layer using equation 15.

10. Step 9: After getting the value of the derived parameter and, calculate the weight change parameters in the hidden layer with SGD method using equation 16 and equation 17. Perform calculations to the hidden layer first.

11. Step 10: If the termination condition is not met, i.e. if epoch \( \neq \) maximum epoch, repeat the calculation starting from step 1.
2.5. DNN testing
The testing process is done by classifying test data, then the outputs are compared with the actual target. The steps for DNN testing are as follows [16]:
1. Step 1: Enter all input values of \( X \).
2. Step 2: Determine the weight value using the final weight value obtained from the training results.
3. Step 3: Perform forward propagation phase calculations, i.e. calculating the output value in the hidden layer using equation 5 and equation 6 or equation 7. Calculate the output value at the output layer using equation 8 and equation 9. Generate an error value from the test data by calculating using equation 10.
Step 4: With a threshold do a threshold on the final result of the error value. The threshold used is 0.5. If \( y > 0.5 \) then the output value is 1 and vice versa, if \( y \leq 0.5 \) then the output value is 0.

2.6. Evaluation
The evaluation stage calculates the accuracy value from the classification results. The evaluation is carried out by comparing the accuracy values of each normalization method.

3. Results
This research tests several scenarios to obtain the best value of accuracy with the DNN method. Scenario 1 aims to test the best accuracy value of the liver cancer dataset with several different activation functions. Scenario 2 aims to test the best accuracy value of the liver cancer with Min-max normalized dataset with several different activation functions. Scenario 3 aims to test the best accuracy value of the liver cancer with Sigmoid normalized dataset with several different activation functions. Scenario 4 aims to test the best accuracy value of the liver cancer with Softmax normalized dataset with several different activation functions. Scenario 5 compares the entire accuracy values of each scenario to get the best accuracy value from the DNN method.

The dataset used in this study consisted of 600 liver cancer data divided into 300 data on liver cancer class and 300 data on normal class with a total of 328 MicroRNA features. Testing is done using learning rate parameter (0.1, 0.2, 0.3, 0.4, 0.5), hidden layer (2, 3, 4), and 200 epoch. The testing scenarios can be seen in Table 1.

| Scenario | Dataset                      | Activation Function | Experiment |
|----------|------------------------------|--------------------|------------|
| 1        | Liver cancer dataset         | ReLU               | 1          |
|          |                              | Sigmoid            | 2          |
|          |                              | TanH               | 3          |
| 2        | Min-max normalization dataset| ReLU               | 4          |
|          |                              | Sigmoid            | 5          |
|          |                              | TanH               | 6          |
| 3        | Sigmoid normalization dataset| ReLU               | 7          |
|          |                              | Sigmoid            | 8          |
|          |                              | TanH               | 9          |
| 4        | Softmax normalization dataset| ReLU               | 10         |
|          |                              | Sigmoid            | 11         |
|          |                              | TanH               | 12         |
| 5        | Scenario 1 – 4 datasets      | ReLU               | 1 - 12     |
|          |                              | Sigmoid            |            |
|          |                              | TanH               |            |

3.1. Scenario 1 results
Scenario 1 tests the accuracy of the liver cancer dataset without data normalization. The results of scenario 1 testing can be seen in the following Table 2.
Table 2 Scenario 1 results

| Experiment | Learning Rate | Hidden Layer | Accuracy |
|------------|---------------|--------------|----------|
| 1          | 0.05          | 3            | 95 %     |
| 2          | 0.05          | 2            | 93.89 %  |
| 3          | 0.04          | 2            | 93.33 %  |

In the scenario 1, the highest accuracy value is in the results of experiment 1, namely the classification of liver cancer without data normalization with the ReLU activation function using 3 hidden layers and 0.05 learning rate with an accuracy value of 95%.

3.2. Scenario 2 results
Scenario 2 tests the accuracy of the liver cancer dataset with Min-Max data normalization. The results of scenario 2 testing can be seen in the following Table 3.

Table 3 Scenario 2 results

| Experiment | Learning Rate | Hidden Layer | Accuracy |
|------------|---------------|--------------|----------|
| 4          | 0.04          | 2            | 98.33 %  |
| 5          | 0.05          | 2            | 97.22 %  |
| 6          | 0.04          | 2            | 97.78 %  |

In the scenario 2, the highest accuracy value is in the results of experiment 4, namely liver cancer classification with Min-Max data normalization and ReLU activation function using 2 hidden layers and 0.04 learning rate with an accuracy value of 98.33%.

3.3. Scenario 3 results
Scenario 3 tests the accuracy of the liver cancer dataset with Sigmoid data normalization. The results of scenario 3 testing can be seen in the following Table 4.

Table 4 Scenario 3 results

| Experiment | Learning Rate | Hidden Layer | Accuracy |
|------------|---------------|--------------|----------|
| 7          | 0.05          | 3            | 96.67 %  |
| 8          | 0.05          | 2            | 96.11 %  |
| 9          | 0.05          | 2            | 95.56 %  |

In the scenario 3, the highest accuracy value is in the results of experiment 7, namely liver cancer classification with Sigmoid data normalization and ReLU activation function using 3 hidden layers and 0.05 learning rate with an accuracy value of 96.67%.

3.4. Scenario 4 results
Scenario 4 tests the accuracy of the liver cancer dataset with Softmax data normalization. The results of scenario 4 testing can be seen in the following Table 5.

Table 5 Scenario 4 results

| Experiment | Learning Rate | Hidden Layer | Accuracy |
|------------|---------------|--------------|----------|
| 10         | 0.05          | 2            | 97.22 %  |
| 11         | 0.05          | 2            | 94.44 %  |
| 12         | 0.05          | 2            | 96.67 %  |
In the scenario 4, the highest accuracy value is in the results of experiment 10, namely liver cancer classification with Softmax data normalization and ReLU activation function using 2 hidden layers and 0.05 learning rate with an accuracy value of 97.22%.

3.5. Scenario 5 results
Scenario 5 compares the accuracy values of each scenario to get the best accuracy value. The results of scenario 5 testing can be seen in the following Table 6.

Table 6 Scenario 5 results

| Scenario | Experiment | Learning Rate | Hidden Layer | Accuracy  |
|----------|------------|---------------|--------------|-----------|
| 1        | 1          | 0.05          | 3            | 95 %      |
|          | 2          | 0.05          | 2            | 93.89 %   |
|          | 3          | 0.04          | 2            | 93.33 %   |
| 2        | 4          | 0.04          | 2            | 98.33 %   |
|          | 5          | 0.05          | 2            | 97.22 %   |
|          | 6          | 0.04          | 2            | 97.78 %   |
| 3        | 7          | 0.05          | 3            | 96.67 %   |
|          | 8          | 0.05          | 2            | 96.11 %   |
|          | 9          | 0.05          | 2            | 95.56 %   |
| 4        | 10         | 0.05          | 2            | 97.22 %   |
|          | 11         | 0.05          | 2            | 94.44 %   |
|          | 12         | 0.05          | 2            | 96.67 %   |

The comparison of DNN accuracy values for each scenario is shown by a comparison chart of the DNN scenario results which can be seen in Figure 2 below. Figure 2 shows that the best accuracy value is in experiment 4 with an accuracy value of 98.33%. The best parameters are obtained by using 2 hidden layers, 0.04 learning rate and 200 epochs. The test results also show that the data normalization with the Min-Max normalization method has better accuracy than Sigmoid and Softmax, and the ReLU activation function has a better accuracy than the Sigmoid and TanH activation functions.

As seen in Figure 2, the ReLU activation function has a better accuracy than the other activation functions in all scenarios. The ReLU activation function after it was introduced is a better substitute for the Sigmoid function because it has a better impact on different Machine Learning tasks [17]. Sigmoid and Softmax activation functions generally work well in the case of classification, but are sometimes avoided due to Vanishing Gradient Problems.

Vanishing Gradient Problems occur because gradient values or parameter changes are very small or close to zero [18]. In this research, Experiment 2 experienced a vanishing (loss) gradient because the derivative value of the Sigmoid function that occurred when backward propagation was very close to zero. In contrast to Experiment 5, Experiment 8, and Experiment 11, whose derivative values are more stable because they use smaller values of $x$ over a smaller range [0-1] which results in the derivative value of the Sigmoid function is also being more stable. The same case also occurred in Experiment 3 where the Softmax function – which is also a development of the Sigmoid function - where the derived value also loses the gradient.
To solve or handle the Vanishing Gradient Problems is still a topic of research among experts. Some of the solutions that can be developed are Long-Short Term Memory (LSTM) [19], Gated Recurrent Neural Network (GRN) [20], or Recurrent Identity Network (RIN) [21].

According to research by (Ertuğrul, 2018) the reasons behind the lower level of accuracy are (1) because it uses a different activation function at each hidden layer and (2) the neuron activation function at the hidden layer is trained according to the number of inputs of a particular neuron and the desired output [22]. In the research by (Ertuğrul, 2018) it was also found that by applying the activation function to the first neuron in the hidden layer, the level of accuracy obtained tended to be lower than the activation function that was trained on all the input and output features desired and used on each neuron [22].

4. Conclusion
The conclusion of this research is the data normalization method and different activation functions can affect the accuracy value of the Deep Neural Network (DNN) method. The DDN method using the Min-Max data normalization and the ReLU activation function produces the best accuracy value in the classification of liver cancer with MicroRNA data. The accuracy value obtained was 98.33% with parameter of 2 hidden layer, 0.04 learning rate, and 200 epochs.

References
[1] Noviani E, Sidarto K A and Putra Y S 2012 Pengelompokan Pasien Kanker Liver Berdasarkan Data Ekspresi Gen Dengan Struktur Papan Catur vol. XVI 1053-1061
[2] I. A. f. R. o. C. (IARC) 2018 GLOBOCAN 2018: Incidence, Mortality and Prevalence by cancer site, [Online Available: http://globocan.iarc.fr/Pages/fact_sheets_population.aspx. [Accessed 3 September 2019].
[3] KKRI 2015 Buletin Jendela Data & Informasi Kesehatan [Online Available: http://www.depkes.go.id/download.php?file=download/pusdatin/buletin/. [Accessed 13 Agustus 2019].
[4] Lan H, Lu H, Wang X and Jin H 2015 BioMed Research International 1-17
[5] Madhavan D, Cuk K, Burwinkel B and Yang R 2013 Frontiers in Genetics 4(116) 1-13
[6] Ahmad A 2017 Jurnal Teknologi Indonesia 1-5
[7] Spencer M, Eickholt J and Cheng J 2015 IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 12(1) 103-112
[8] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Journal of Machine Learning Research 15 1929-1958
[9] Thomas J and Sael L 2017 Deep Neural Network Based Precursor microRNA Prediction on Eleven Species arXiv preprint arXiv:1704.03834
[10] Joshi A and Metha A 2018 International Journal of Recent Scientific Research 9(4D) 788-792
[11] Wibowo A and Ningrum A C 2019 Perbandingan Nilai Akurasi Klasifikasi Kanker Payudara Dengan Data MicroRNA Menggunakan Metode Deep Neural Network (DNN) Tanpa dan Dengan Optimasi Batch Normalization (Semarang: Indonesia)
[12] Chamidah N, Wiharto and Salamah U 2012 ITSMART 1(1) 28-33
[13] Pedamonti D 2018 Comparison of non-linear activation functions for deep neural networks on MNIST classification task 1 1-5
[14] Jain Y K and Bhandare S K 2011 International Journal of Computer & communication Technology 2(8) 45-50
[15] Goodfellow I, Bengio Y and Courville A 2017 Deep Learning (Cambridge: Massachusetts: MIT Press)
[16] A Y T Ng, deeplearning.ai 2017 [Online]. Available: https://www.youtube.com/channel/UCcIXc5mJsHVYTZR1maL5I9w/playlists. [Accessed 9 7 2019].
[17] Nair V and Hinton G E 2010 Rectified Linear Units Improve Restricted Boltzmann Machines in Proceedings of the 27th International Conference on Machine Learning, Halfa, Israel
[18] Putra J W G 2018 Pengenalan Konsep Pembelajaran Mesin dan Deep Learning (Tokyo: Japan)
[19] Fischer T and Krauss C 2017 European Journal of Operational Research, no. https://doi.org/10.1016/j.ejor.2017.11.054, pp. 654-669
[20] Chung J, Gulcehre C, Cho K H and Bengio Y 2014 Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling vol. I, no. arXiv:1412.3555v1 [cs.NE] pp. 1-9
[21] Hu Y, Huber A, Anumula J and Liu S C 2019 Overcoming The Vanishing Gradient Problem in Plain Recurrent Networks vol. III, no. arXiv:1801.06105v3 [cs.NE] pp. 1-20
[22] Ertuğrul O F 2018 Neural Networks 99 148-157