Optimization of neural network for cancer microRNA biomarkers classification

1A Wibowo, 2P W Wiryawan, and 3N I Nuqoyati

1,2,3Department of Informatics, Faculty of Sciences and Mathematics, Diponegoro University
Jl. Prof Soedarto, SH, Ngesrep, Semarang 50275, Indonesia
E-mail: bowo.adi@live.undip.ac.id

Abstract. Cancer is still a significant problem for people today because it is one of the biggest causes of death in the world. Based on GLOBOCAN data in 2018, breast cancer accounted for the world's largest cancer mortality rate in women by 6.6% with total deaths amounting to 626,679 from 2,088,849 cases of cancer in the world. The high mortality rate of breast cancer is caused by the lack of effective early detection of the disease. MicroRNAs play an essential role in regulating cell division cycles, apoptosis, senescence, migration and cell invasion, and metastasis. The expression of microRNA in breast cancer shows a pattern compared to normal breasts, thus indicating its role as a potential diagnostic marker. Cancer classification using microRNA as a feature has been done in previous studies using Neural Network Backpropagation, however without optimization and tuning parameters. In this paper, we investigated the best optimization algorithm and tuning parameter of neural network backpropagation for cancer classification using microRNA feature. The optimization algorithms were Gradient Descent, Momentum, AdaGrad, AdaDelta, RMSProp, and Adam. The result of the experiment showed that Adam and RMSPop optimizer produced high accuracy which reached 98.536% and 98.54762% accuracy.

1. Introduction
Cancer is the leading cause of death in the world. In 2018, around 9.5 million deaths were caused by cancer [1]. Cancer is a type of disease that arises due to abnormal growth of body tissue cells that turn into cancer cells [1]. Various efforts were made to prevent cancer before it happens or what is called as early detection of cancer. It is a medical effort to increase awareness of cancer. With the early detection of cancer, the life expectancy of a person affected by cancer will increase. One way to do this early detection of cancer is using microRNA or commonly called miRNA. MicroRNA is a non-coding RNA that helps translate genetic DNA information into proteins [2]. MicroRNA is transcribed from DNA but is not processed into proteins or polynucleotides, so it is said to be produced by a non-coded part of DNA. MicroRNA regulates target genes either by microRNA degradation or by repression translation. Each microRNA can regulate up to hundreds of target genes. This microRNA can be used for early detection of cancer by using machine learning.

Cancer detection using microRNA as a feature has been done in previous studies [3]. In this study, the search for the best features and exploration of machine learning methods was produced and showed that the best algorithm was KNNE. The next research showed that deep learning is the best algorithm for microRNA data classification, but without optimization and tuning parameters [4]. Research related to the selection of the best microRNA features has been done with co-training, self-learning, DBN and active learning [5-6]. Then in other studies, microRNA features were chosen for the classification of...
ovarian cancer [7] and breast cancer [8] using backpropagation neural networks. However, there is no research on selecting the right neural network optimization function used for microRNA data.

Gradient Descent is the most common optimization performed on Backpropagation. Research related to Gradient Descent has been done [9-12]. Other optimizations such as momentum [13], AdaGrad [14], RMSProp [15], AdaDelta [16], and Adam [17] also has been done by researchers recently because of their high effectiveness.

In this study, some neural network optimization methods were compared to find out the best optimization to be the classification method for cancer and how neural network parameters can influence the accuracy value. Furthermore, the results obtained will be analyzed.

2. Literature and Method Review

2.1. Backpropagation

The artificial neural network is a computational model that mimic the process of neural networks in living things. The artificial neural network has some characteristics, (1) each neuron is connected, (2) a method to determine the weight value for each connection and, (3) activation function [18]. The model built in this study is the neural network with the feedforward algorithm and uses backpropagation in training. The architecture of this neural network model using 3 layers, namely the input layer, hidden layer, and output layer as shown in Figure 1.

**Figure 1.** Backpropagation Neural Network Architecture

In the feedforward phase, the data will be processed forward from the input to the hidden layer and then to the output layer. In the backpropagation phase, the process will repeatedly run from output to input; then the network will automatically minimize the errors. The error calculation used here uses sigmoid cross entropy.

2.2. Optimizer

The optimizer in the neural network serves to optimize the network so that better result can be obtained. Optimization techniques that are often used are Gradient Descent [19], momentum [13], AdaGrad [14], RMSProp [15], AdaDelta [16], and Adam [17].

2.2.1. Gradient Descent

Gradient Descent is the most common optimization technique used in neural networks and is the basis for the development of other optimization techniques. Gradient descent aims to find the minimum value of a function. Unfortunately, gradient descent has a disadvantage when the slope of the curve is too sharp and different from one another [20].

2.2.2. Momentum Optimizer

Momentum optimizer is a modification of gradient descent, this optimizer accelerates the convergence and reduces the number of oscillations. In the momentum optimizer, the first thing to do is to add the $\gamma$ component to the update parameter in the calculation of weights as in equation (1) and equation (2).
The index $\gamma$ shows the parameters of the range $(0, 1)$.

2.2.3. AdaGrad Optimizer

AdaGrad was introduced in 2011 [14]. AdaGrad is a gradient-based algorithm. AdaGrad is an optimization method that has advantages in adapting the learning rate according to the value of the existing parameters. A great learning rate value will be used for parameters that rarely appear and small learning rate value for parameters that often appear. The disadvantage of AdaGrad is that the decrease in the learning rate can reach 0. The AdaGrad formula is as in equation (3) and equation (4).

$$G^k = G^{k-1} + (\theta^{k-1})^2$$

$$\theta^k = \theta^{k-1} - \frac{\alpha}{\sqrt{G^k} + \epsilon} \cdot \nabla f(\theta^{k-1})$$

Where $\cdot$ and "sqrt" are element-wise operations.

2.2.4. AdaDelta Optimizer

AdaDelta is the development of AdaGrad which aims to correct the problem at AdaGrad's learning rate which continuously decreases in each iteration. The main idea of AdaDelta is the adaptation of learning rate based on gradient history but only takes from the nearest time, not from all current history like AdaGrad. Moreover, the second idea is that AdaDelta uses all components that provide acceleration conditions, which accumulate the history updates (similar to momentum).

The steps of update in AdaDelta are as follows:

1. Calculate the gradient $g_t$ at time $t$.
2. Calculate the gradient as in equation (5).
$$E[g_t^2]_t = \rho E[g_t^2]_{t-1} + (1 - \rho) g_t^2$$
3. Calculate updates as in equation (6).
$$\Delta x_t = -\frac{\sqrt{E[\Delta x_t^2]_{t-1} + \epsilon}}{\sqrt{E[g_t^2]_{t-1} + \epsilon}} g_t$$
4. Accumulate update values (same steps as momentum) as in equation (7).
$$E[\Delta x_t^2]_t = \rho E[\Delta x_t^2]_{t-1} + (1 - \rho) \Delta x_t^2$$
5. Apply the update as in equation (8).
$$x_{t+1} = x_t + \Delta x_t$$

Where $\rho$ is decay constant and $\epsilon$ is used as a numerical stabilizer (which is usually a very small number).

2.2.5. RMSProp Optimizer

RMSProp is unpublished adaptive learning proposed by Geoff Hinton in Lecture 6e at Coursera Class [15]. RMSProp was developed to overcome AdaGrad's learning rate which continuously decreases in each iteration. RMSProp is identical to AdaDelta but with a slight difference in the calculation of updates as in equation (9).

$$E[g_t^2]_t = \rho E[g_t^2]_{t-1} + (1 - \rho) g_t^2$$

$$x_{t+1} = x_t - \frac{\eta}{\sqrt{E[g_t^2]_{t-1} + \epsilon}} g_t$$

Where the value of $\rho$ proposed is 0.9 and the value of $\eta$ is 0.001.

2.2.6. Adam Optimizer
Adaptive moment estimation, or commonly called Adam, is an optimization that is widely used in neural network models, especially for large datasets and high dimensional parameters. Like AdaDelta and RMSProp, Adam adapted the learning rate according to parameters. In addition to adapting parameters based on the mean of old gradient $m_t$, Adam also adapted the parameters based on the mean of the old gradient $v_t$.

The steps of update at Adam are as follows:
1. Calculate the gradient $g_t$ at time $t$.
2. Update the first bias estimation as in equation (11).
   \[ m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t \]  \hspace{1cm} (11)
3. Update the second bias estimation as in equation (12).
   \[ v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2 \]  \hspace{1cm} (12)
4. Calculate the first bias estimation that has been corrected as in equation (13).
   \[ \hat{m}_t = \frac{m_t}{1 - \beta_1^t} \]  \hspace{1cm} (13)
5. Calculate the second bias estimation that has been corrected as in equation (14).
   \[ \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \]  \hspace{1cm} (14)
6. Update the parameters as in equation (15).
   \[ \theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \]  \hspace{1cm} (15)

2.3. Evaluation Method
The evaluation used to test this experiment is accuracy, sensitivity, and precision. First, we calculated the value of true positive, true negative, false positive, and false negative. True positive is the number of data that is correctly identified as positive class. The true negative is the number of data that is correctly identified as the negative class. False positive is the number of data that is incorrectly identified as positive class. False negative is the number of data that is incorrectly identified as the negative class. Usually, the confusion matrix is used, as in Table 1.

| Predicted Class | Targeted Class |
|-----------------|----------------|
| Positive        | Positive       |
|                 | True Positive  |
|                 | False Positive |
| Negative        | False Negative |
|                 | True Negative  |

1. Accuracy is a value that shows how similar value is predicted to the actual value. The formula is as in equation (16).
   \[ Accuracy = \frac{\Sigma True \ positive + \Sigma True \ negative}{\Sigma Total \ Population} \]  \hspace{1cm} (16)
2. Sensitivity is a value that shows the probability of detection or is a value that shows how many positive conditions are correctly detected as positive. The formula is as in equation (17).
   \[ Sensitivity = \frac{\Sigma True \ positive}{\Sigma Condition \ positive} \]  \hspace{1cm} (17)
3. Precision is a value that indicates the level of accuracy between the detection results requested by the user and the answers given by the system. The formula is as in equation (18).
   \[ Precision = \frac{\Sigma True \ positive}{\Sigma True \ Positive + \Sigma False \ Positive} \]  \hspace{1cm} (18)
3. Experiment and Result

3.1. Dataset
The dataset used was taken from the National Center for Commons Genomic Data Institute which can be accessed at http://gdc.cancer.gov/. The data taken were normal and tumor data (two classes) with a total of 208 data with 104 data for normal class and 104 data for cancer class. The dataset consisted of 17 different microRNA expressions as features. These 17 features were selected based on the paper by Lan et al. [21].

3.2. Implementations
The neural network algorithm was implemented using TensorFlow and Scikit-Learn Library. Existing data were separated using stratified cross-validation with the number $K = 5$. The experiment was carried out using Intel Core i7 2.20 G.Hz and Nvidia Geforce 740M GPU.

3.3. Scenario I
The scenario I aimed to determine the best architecture of the Neural Network (the number of epochs, learning rate, and some neurons) with the primary optimization function of Gradient Descent. The results obtained from the scenario I are shown in Table 2, Table 3, and Table 4.

The first step in the scenario I was to determine the number of neurons with the initial learning rate 0.01 and the number of epochs 50. The number of neurons starts from 3 to the same as the number of feature data. The result in Table 2 showed that the best number of neurons is nine according to the rules proposed by the Masters [22] that for networks consisting of three layers with n input and the hidden layer m output is $\sqrt{n \times m}$. The result that has been obtained will be used as the basis for the second step in scenario I.

The second step in the scenario I was to find the learning rate value by using the number of neurons from the previous step which was 9 and using 50 epoch. The value of the learning rate tested ranged from 0.1, 0.05, 0.03, 0.01, 0.001. The result obtained can be seen in Table 3 where the learning rate of 0.1 was the best result.

The third step in the scenario I was to find the best epoch value by using the value of the number of neurons and learning rate from step 1 and step 2. Table 4 shows the result of the third step with the best result of 175 epochs.

3.4. Scenario II
Scenario II aimed to determine the best optimizer function for microRNA data on cancer classification. The optimizer function used were by Table 5, namely AdaDelta, Adam, Gradient Descent, Momentum, and RMSProp. Each of these optimizers has an activation function which their combination produces the most optimum result, so we did not experiment to try each of activation function to each optimizer. The optimizer function and its activation function is showed in Table 5.

The architecture used referred to the scenario I experiment, using the number of neurons 9, epoch 175, and also the value of learning rate 0.1. In the scenario II model experiment, the test was performed again by replacing the optimizer function and adjusting the activation function according to the optimizer. Table 5 shows each optimizer and its activation function. Adam's experiment used the default bias values of 0.9 and 0.999 and epsilon values of $10^{-8}$ [17]. The overall results of the best comparison of accuracy, sensitivity, and precision are shown in Table 6. The results showed that Adam and RMSProp were optimizers with the best accuracy values. Adam optimizers produced a better accuracy since it was designed to combine the advantages of two recently popular methods, AdaGrad and RMSProp, Adam worked well with sparse-gradients, on-line and non-stationary settings. Adam did not require a stationary object and it naturally performed a form of step size annealing [17]. While RMSProp algorithm can speed up the learning process or at least not slow it down in the horizontal direction [15]. Evaluation results of all models are visualized by Figure 2.

Table 2. The evaluation results of the first model

|   | 17 | 81.70383 | 63.52381 | 80 |
|---|---|---------|---------|---|
### Table 3. The evaluation results of the second model

| Learning Rate | Accuracy  | Sensitivity | Precision |
|---------------|-----------|-------------|-----------|
| 0.1           | 96.59582  | 99.04762    | 94.49464  |
| 0.05          | 95.70266  | 95.23809    | 96.59091  |
| 0.03          | 96.13182  | 99.04762    | 94.04762  |
| 0.01          | 95.60743  | 96.99999    | 94.62919  |
| 0.001         | 50.24390  | 100         | 50.24390  |

### Table 4. The evaluation results of the third model

| Epoch | Accuracy  | Sensitivity | Precision |
|-------|-----------|-------------|-----------|
| 50    | 92.23867  | 98.09524    | 88.54761  |
| 75    | 80.50173  | 99.04762    | 72.41270  |
| 100   | 90.82289  | 83.71428    | 97.88888  |
| 150   | 97.54820  | 98.09524    | 97.27273  |
| 175   | 98.02381  | 98.04762    | 98.04762  |
| 200   | 97.04820  | 96.04762    | 98.13852  |
Figure 2. Graph of evaluation results of all models

Table 5. Activation function for each optimizer

| Optimizer         | Activation Function |
|-------------------|---------------------|
| AdaDelta          | Sigmoid             |
| Adam              | RELU                |
| Gradient Descent  | Sigmoid             |
| Momentum          | Sigmoid             |
| RMSProp           | RELU                |

Table 6. Comparisons of evaluation results

| Optimizer         | Accuracy  | Sensitivity | Precision |
|-------------------|-----------|-------------|-----------|
| AdaDelta          | 50.243    | 100         | 50.2439   |
| Adam              | 98.536    | 98.09524    | 99.04762  |
| Gradient Descent  | 94.64286  | 98.04762    | 91.92923  |
| RMSProp           | 98.54762  | 98.99999    | 98.18182  |
| Momentum          | 98.07143  | 96.14285    | 100       |

4. Conclusion
In this paper, we showed a study of the best activation function in the neural network training process used for cancer classification using microRNA data based on maximum accuracy value. The optimizer algorithms were Gradient Descent, Momentum, AdaGrad, AdaDelta, RMSProp, and Adam. The results of the experiments in scenario II showed that Adam and RMSProp produced the best accuracy, Adam achieved an accuracy value of 98.536% and RMSProp produced an accuracy value of 98.54762%. By this result, we hope it can gain improvement in the establishment of better and faster early cancer detection technology.

Acknowledgment
This work was supported by the RISTEKDIKTI, The Republic of Indonesia for PASCA DOCTOR Research Grant 2018.

References
[1] IARC. Fact Sheets by Cancer. 2018. Retrieved November 8, 2018, from http://gco.iarc.fr/today/data/factsheets/cancers/20-Breast-fact-sheet.pdf

[2] N. Rosenfeld, R. Aharonov, E. Meiri dan S. Rosenwald, MicroRNAs accurately identify cancer tissue origin, 2008.

[3] Kim K and Cho S 2010 International Conference on Neural Information Processing (ICONIP) 2 234.

[4] Waspada I, Wibowo A, Meraz S 2017 Jurnal Ilmu Komputer dan Informasi 10 108-15

[5] Ibrahim R, Yousri N, Ismail M and El-Makky N 2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).

[6] Ibrahim R, Yousri N, Ismail M and El-Makky N 2014 Proc. IEEE Int. Conf. Eng. Med. Biol. Soc. pp. 3957.

[7] Anjarwati, Wibowo A, Adhy S, Kusumaningrum R 2018 Journal of Physics: Conference Series 2018 1025 012090

[8] Khasburrahman K, Wibowo A, Waspada I, Hashim HB, Jatmiko W 2017 In Informatics and Computational Sciences (ICICoS), (Indonesia: IEEE) Nov 15 pp. 165

[9] Breuel, M 2015. On the Convergence of SGD Training of Neural Networks ArXiv Preprint arXiv: 1508.02790

[10] Hardt M, Recht B, and Singer Y 2015 Train faster. generalize better: Stability of stochastic gradient descent. ArXiv Preprint.

[11] Mandt S, Hoffman D and Blei D 2016 A Variational Analysis of Stochastic Gradient Algorithms ArXiv Preprint.

[12] Neyshabur B, Salakhutdinov R, and Srebro N 2015 In Advanced in neural information processing system.

[13] Qian N 1999 Neural Networks : The Official Journal of the International Neural Network Society 12 145

[14] Duchi J, Hazan E and Singer Y 2011 The Journal of Machine Learning Research 12 2121

[15] Tieleman T and Hinton E 2012 Lecture 6.5 - rmsprop, COURSERA: Neural networks for machine learning.

[16] Zeiler M 2012 ADADELTA: An Adaptive Learning Rate Method. ArXiv Preprint arXiv:1212.5701.

[17] Kingma D and Ba J 2015 Adam: A Method for Stochastic Optimization. In Proceedings of international conference on learning representations.

[18] Siang J 2005 Jaringan Syaraf Tiruan dan Pemrograman Menggunakan Matlab (Yogyakarta: publisher Andi).

[19] LeCun Y, Haffner P, Bottou L, Bengio Y 1999 Object recognition with gradient-based learning. In Shape, contour and grouping in computer vision pp. 319 (Springer, Berlin, Heidelberg)

[20] Sutton R 1986 Two problems with backpropagation and other steepest-descent learning procedures for networks Proc. 8th Annual Conf. Cognitive Science Society.

[21] Lan H, Lu H, Wang X, Jin H 2015. MicroRNAs as potential biomarkers in cancer: opportunities and challenges. BioMed Research International.

[22] Masters T 1993 Practical neural network receips in c++. Morgan Kauf.amanho.