**Using Genetic Algorithm-based Feature Selection and Shared Weights Autoencoder in Neural Network for Breast Cancer Analysis**

Author Name (Paper 195 assignment 2)

Research School of Computer Science, Australian National University [u1234567@anu.edu.au](mailto:u1234567@anu.edu.au)

**Abstract.** As the base for breast cancer analysis, cell information such as cell size, shape and texture features has been widely used. Traditional machine learning methods are often used to analyze the cell information and classify the cell type. In this paper, two new neural network models will be presented for this classification task. One is the combination of artificial neural network(ANN) and shared weights autoencoder, and the other one contains both ANN and genetic algorithm(GA)-based feature selection. In these models, the ANN is used for classification. Shared weights autoencoder and GA-based feature selection are used to solve the redundant features problem. Furthermore, I conduct several experiments with the Breast Cancer Wisconsin(Diagnostic) dataset that has well collected data. My experiments investigate the impact of autoencoder, the effectiveness of shared weights technique and the impact of GA-based feature selection. I also analyze the performance of my two models and logistic regression model. My results confirm the feature extraction ability of autoencoder and the effectiveness of shared weights technique. What is more, the results show the superior performance of my two models compared with logistic regression model. And GA-based feature selection performs better than shared weights autoencoder but cost more computation resources.

**Keywords:** Neural network, Shared weights, Autoencoder, Genetic algorithm, Breast cancer, Feature extraction, Feature selection

# Introduction

As an essential part in both cancer treatment and cancer prevention, cancer prediction is an approach using different bio- information to predict whether a cell is malignant. With the aid of computers, microscope image can be converted to accurate feature information, which can highly improve prediction accuracy and objective feature assessment [6].

Cancer prediction with cell information can be regarded as a simple classification task in machine learning area. In general, a classification task is using some features of an object to tell which category it belongs to. There are many methods to implement this task, such as logistic regression [1], C4.5 decision trees [4] and Naïve Bayesian learning [5].

In the previous work, different methods have been investigated on this cancer prediction task. W. H. Wolberg et al use logistic regression and inductive machine learning method, which provide the base for breast cell analysis [1]. G. R. M.

A. Sizilio et al use fuzzy system, which can reduce the variation hit of malignant cases [7]. Y. M. George et al test the performance of SVM, which can create a soft margin that permits some misclassification [8].

Traditional machine learning methods have shown a good performance in the breast cancer classification task, for example, logistic regression method gains the accuracy with the value of 96.2% [1]. However, problems exist in traditional methods drive the design of two models used in the experiments. One problem is that traditional methods cannot describe the non-linear relation between independent and dependent variables [2]. Because the neural network can detect all the possible relations between variables [9, 10]. Neural network method can be used in classification task to solve the problem stated above. The other problem is that input features have large dimensions. Some redundant features may impact the performance of the neural network and lead the network to overfit on the old examples but perform badly in the new data [3]. Autoencoder can be used for feature extraction in this task. An autoencoder is an artificial neural network that uses unsupervised learning to reduce data dimension, which can be used for feature extraction. Good autoencoder algorithm can also avoid noise pattern and peaking of weights [12]. Besides, inspired by the shared weights auto-associative network in image compression area, I use the shared weights technique in the autoencoder to get a better feature extraction result. Since the weights in a shared weights network must be invertible, it will give a better compressed result than the original network [13]. Therefore, ANN and shared weight autoencoder were used in my first model to do classification and feature extraction respectively.

However, based on further investigation, feature extraction method often deletes some discriminative features and

reduce the verification performance [24]. On the other hand, some extracted features may have no effects in the performance improvement [25]. Thus, feature selection method needs to extract useful and discriminative features. Genetic algorithm is a powerful tool for feature selection, especially when the input feature dimension is large [26]. By reducing the feature dimensions, genetic algorithm can not only reduce the computation complexity but also improve the performance of classifier in classification task [27]. Therefore, the second model contains both GA-based feature selection and ANN with possibility of potential improvement.

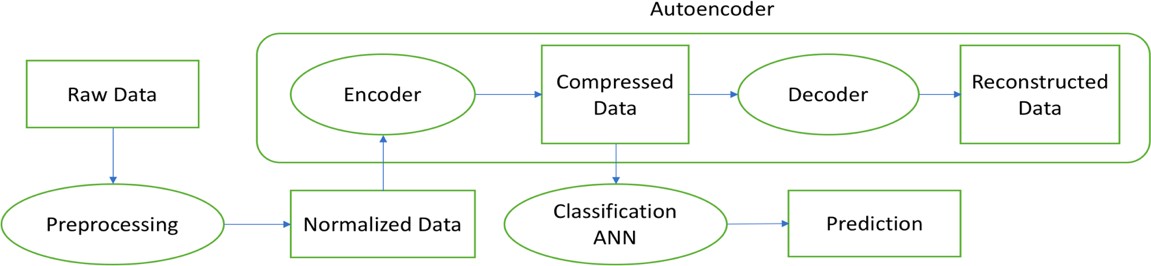
The structure of this paper is organized as following. Section 2 describes the methods used in my two models. Section 3 summarizes the experimental results and discussions. Conclusions and the direction for future work are given in Section 4.

# Methodologies

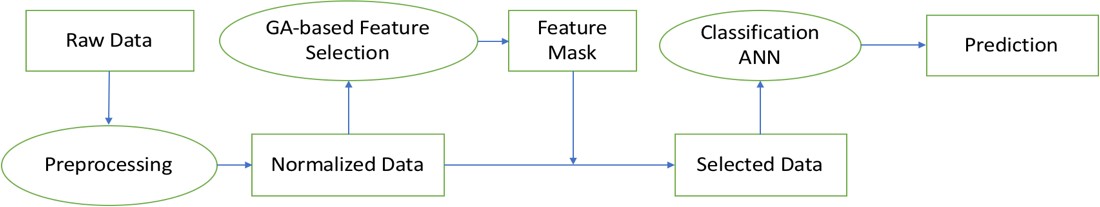
I formulate the task of distinguishing the breast cell from benign or malignant as a classification problem and design two neural network architectures for this problem. This first model contains ANN and shared weights autoencoder. The aim of this model is to investigate the performance of neural network, the impact of autoencoder and the effectiveness of shared weights technique. The second model contains ANN and GA-based feature selection. This model is designed to investigate the impact of GA-based feature selection. The Breast Cancer Wisconsin (Diagnostic) dataset is chosen as the dataset for training (section 2.1). The overview of the two network models is shown in Fig.1 and Fig.2 respectively.

As shown in the Fig.1, the first model can be divided in three parts: 1) data preprocessing for dealing with the input data (section 2.2). 2) a shared weights autoencoder for extracting the important features (section 2.3). 3) artificial neural network for classification (section 2.4). The Fig.2 shows the structure of the second model. The input preprocessing and classification parts are the same as the first model. To select important features, this model uses a GA-based feature selection method instead of autoencoder (section 2.5).

The final output of my models is a probability vector for benign and malignant cell (section 2.4). In the classification ANN training, 10-fold validation and mini-batch were used to make best use of the finite dataset and get an optimal training performance (section 2.6). Two different loss functions were taken for the autoencoder and classification ANN respectively (section 2.5). Both accuracy and F1 score were chosen as the measures for the evaluation (section 2.7).



**Fig. 1.** Overview of the model that contains shared weights autoencoder and ANN.



**Fig. 2.** Overview of the model that contains GA-based feature selection and ANN.

## Dataset Description

In this paper, I choose the Breast Cancer Wisconsin (Diagnostic) dataset with two reasons. Firstly, computer-aided methods have become the best choice in cancer prediction area, and this dataset has been widely used for cancer prediction since 1990s [1,6]. Therefore, this dataset is reliable for cancer prediction task. Secondly, the features in this dataset are well detected. All the features are converted from digital images that are collected from fine needle aspiration (FNA) [1], and there is no missing value.

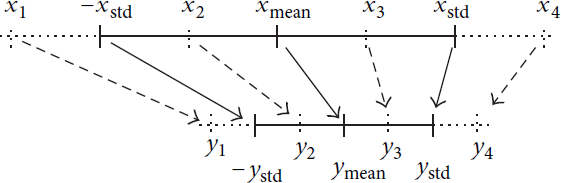
This dataset contains 569 cell samples: 212 are benign and 357 are malignant. Each sample has 32 features, the first feature is the cell id, the second feature is the cell type (malignant or benign). The other features are real-value features computed from each cell nucleus, such as radius, texture, and perimeter.

## Data Preprocessing

In the raw data, most of the features are numerical, while the only non-number feature is the cell label. For the cell label feature, the malignant cell is labelled with “M”, and the benign cell is labelled with “B”. For the ease of processing, cell labels were converted to numbers by replacing “M” with 0 and “B” with 1 and all data were normalized between 0 to 1.

When neural network is used for classification, the data normalization can greatly influence the training process. Sola and Sevilla [14] have shown that the importance of data normalization lies on accelerating the training process and improving the result quality in a neural network model. Jayalakshmi and Santhakumaran [15] also pointed out that normalization can make the feed forward backpropagation more reliable and the diabetes neural network model can give a better result with normalization technique. Within the dataset, the values of some features are small, while other values are large. Therefore, normalized input data was used for the normal distribution method.

Given certain new mean value and deviation value, the normal distribution method can project the original dataset to a new interval (see Fig.3).



**Fig. 3.** Normal distribution method of normalization.

Based on the following formula, the original mean value 𝑥"#$% and distribution value 𝑥&'( can be calculated from the raw data. The 𝑦"#$% and 𝑦&'( are the new mean value and distribution value assigned by us.

y = (,-,./01)∙4567 + ��

(1)

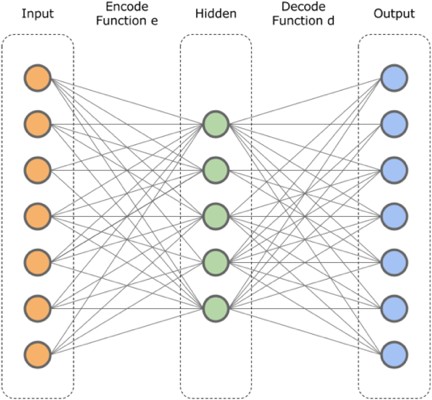
,567

"#$%

The normalized data was used as the input data directly because the normalized data can avoid the peak in weights and reduce the training time. During the training process, the weights can get to the optimal value smoothly with normalized data. There is no need for extra encoding techniques.

## Autoencoder and Shared Weights Technique

In machine learning area, feature extraction is an important approach to reduce the redundant features and select the meaningful features. A meaningful feature representation can highly improve the accuracy for classification task [16]. Instead of using traditional feature selection methods, I use one popular deep learning method, which is autoencoder.

An autoencoder is a multilayer network that can contains an input layer, some hidden layers and an output layer. The aim of the hidden layer is to reconstruct the input data, so the hidden layer contains all the essential information from input layer. As shown in Fig.4, the encoder is this the compress process from input to hidden layer, the decoder is the output layer from hidden layer to output layer. The encoder and decoder are inverse to each other. Thus, the autoencoder architecture is a symmetrical model. After training, the encoder can be used for extracting features from raw data. In my first model, the preprocessed data is used in the autoencoder for data compression.

**Fig. 4.** The structure for autoencoder.

The idea of the shared weights technique comes from the associative neural network in image compression task. The associative neural network is like the standard feed-forward network, expect that the connection weights keep the same between the input layer to hidden layer and hidden layer to output layer. The shared weights associative neural network should have a better performance than normal associative network. Since in a normal network, the first layer is a compressed function and the second layer is a decomposed function. The shared weights network has a stricter constrain, which is first and second layers share the same weight. So, these two functions are invertible, and the network has less free parameters. Intuitively, we can expect that the shared weights network should have a better performance than the normal one. Because the inverse function is “the” function rather than an approximation [13]. Thus, I combine the autoencoder and shared weights technique into shared weights autoencoder, which should have a better performance than the normal autoencoder. Besides, sigmoid activation is used in each layer.

## Classification Artificial Neural Network

In the classification network part, a three-layer artificial neural network was used. It contains input layer, hidden layer and output layer. Each layer contains some nodes. In the input layer, the nodes are called input nodes, and they represent the input variable (compressed features or selected features). In the output layer, each node means a predicted result (probability for belonging cell classes). In the hidden layer, the hidden nodes do not have actual meaning, they are used to model the interconnection between input nodes and the relations between input features and output result.

With labeled data, the ANN can learn the best weights that can describe the relation between inputs and outputs. This process is called training. Backpropagation is used for the training. The main idea for backpropagation is changing the weights in each training to minimize the difference between real label and predicted label [17].

In the classification network, sigmoid function is used in input and hidden layers, and softmax function is used for output layer. The softmax function can constrain the arbitrary value in a vector to range (0,1), and the sum of the values

are 1 [22]. The softmax function is shown in formula (2). In here, z is the hidden value, K is the class number, σ(𝑧); is the output value.

𝑒>?

(2)

σ(𝑧); = B CDE

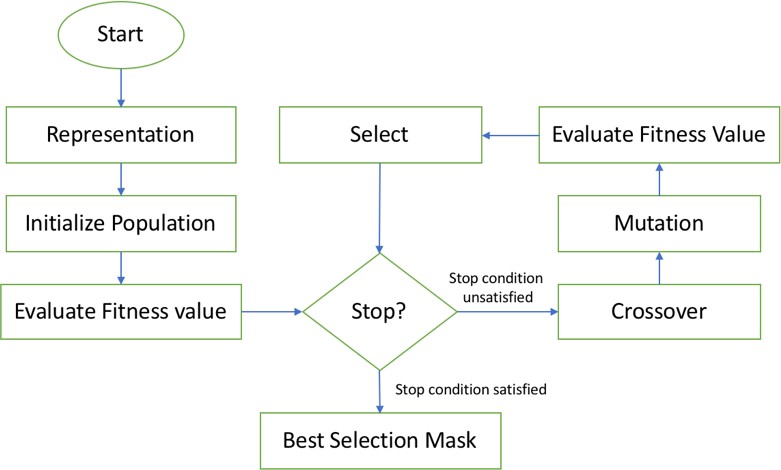
∑

𝑒>A

As for the reasons of choosing softmax as decoder, it is because of the property of softmax function. Firstly, softmax function can map the output value to range (0,1). Secondly, the value can be added up to one. So, after using softmax, the two outputs can represent the probability of being benign and malignant.

## Genetic Algorithm-based Feature selection

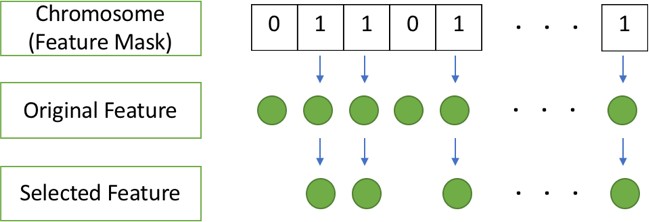
Genetic algorithm, developed by Holland, is a computational optimization procedure inspired from the concept of biological evolution [28]. Genetic algorithm aims to find the best solution in the binary search spaces and manipulates scores of potential solutions [29]. The genetic algorithm simulates the main steps in biological evolution such as reproduction, crossover and mutation, etc. Each possible solution, called a chromosome, is represented by a finite sequence that only contains 0 and 1. The simulated environment, called fitness function, is used to evaluate the quality of a solution. The possibility of survival is related to the fitness value. To select the best features for classifier, I use the genetic algorithm to select informative features and reduce redundant features.



**Fig. 5.** The flow of the GA-based feature selection

The process of GA-based feature selection is shown in Fig.5, and the main concepts are as following.

Representation: In this feature selection task, each chromosome is defined as a mask for features. For instance, “0” means the related feature is not selected, “1” means the related feature is selected. The process of selecting features with feature mask is shown in Fig. 6.



**Fig. 6.** The process of selecting features with feature mask

Fitness function: To evaluate the fitness value of a chromosome, neural network approach is used. The neural network model is the same as the classifier. This process can be divided in three steps: (1) Using the chromosome to get the masked dataset. (2) Using 10-fold validation to train the network model with masked dataset. (3) Calculating the F1 score as the fitness value.

Selection: Proportional selection is used. During the selection, an empty population is initialized and following process is repeated. An individual will be picked from existing population to new population, and the possibility is proportional to the fitness value. When the size of new population is equal to the previous population, this iteration will stop, and the selection is completed.

Crossover: Uniform crossover is used. For each individual, a random value will be generated. If the random value is larger than the crossover rate, this individual will keep the same. Otherwise, this individual will continue the following crossover process. Another individual will be randomly picked from the population. After that, the genes of the new individual will randomly overwrite the corresponding genes of the mutated individual.

Mutation: Each chromosome will be picked from the population sequentially. For each gene in one chromosome, a random value will be generated. If the value is larger than the mutation rate, this gene will keep the same. Otherwise, this gene will change from 0 to 1 or from 1 to 0.

Evolution: The phases from selection to mutation are iterated. For each evolution, the best feature set is recorded, and it was compared with the best feature set in all generations. If the current best feature set is better, it is recorded as the

best feature in all generations. If there is no improvement for 5 times, this genetic algorithm is completed. The best feature set in all stages is the selected feature.

## Loss function

In backpropagation, the loss function calculates the difference between the predicted output and the expected output. Two different loss functions were used for the autoencoder and classification network. For the autoencoder, mean square error (MSE) is used. It can be calculated by the formula (3). The 𝑥F is the input data, the 𝐺(𝑥F) is the predicted label, and 𝑦F is the real label. It can calculate the mean distance between predicted output and real label.

C=E ∑% (𝐺(𝑥 ) − 𝑦 )o

(3)

% FDE F F

For the classification network, cross-entropy method is used. Because in classification task, the cross-entropy has a better performance than square error method [19]. It can be calculated by formula (4). To minimize the cost C, when label 𝑦F is 0, 𝑦F ln 𝐺(𝑥F) is 0, we will try to make 𝐺(𝑥F) ≈ 0, so (1 − 𝑦F)ln (1 − 𝐺(𝑥F)) ≈ 0. When label 𝑦F is 1, vise verse.

C=− E ∑% (𝑦 ln 𝐺(𝑥 ) + (1 − 𝑦 )ln (1 − 𝐺(𝑥 )))

(4)

% FDE F F F F

## 10-fold Cross-Validation and Mini-Batch Method

In the classification ANN and evaluation part of GA-based feature selection, 10-fold cross-validation method is used to split the train and test data. The process is like following. Firstly, the dataset is divided in to 10 equal parts. Secondly one part is chosen for testing, and the other parts are kept for training. After the training and testing, this process in continued until all parts are tested. By 10-fold cross-validation, I can use all data in both training and test process and get an accurate and unbiased result.

What is more, mini-batch gradient decent method is used in classification ANN. The mini-batch method is an algorithm that can split the training dataset into small batches [23]. Each batch is used to calculate the error and update the weights. Using this method, the classifier can get a more robust convergence and avoid the local minimal.

## Evaluation Method

In classification task, it is important to evaluate the model performance. Accuracy and F1 score are used for evaluation.

**Table 1.** Definition for TP, TN, FP and FN

|  |  |  |
| --- | --- | --- |
|  | Class=Yes (Predicted) | Class=No (Predicted) |
| Class=Yes (Actual) | True Positive (TP) | False Negative (FN) |
| Class=No (Actual) | False Positive (FP) | Ture Negative (TN) |

To calculate the F1 score, we have to know the four parameters in table 1: True Positive(TP): the correctly predicted positive number. True Negative(TN): the correctly predicted negative number. False Positive(FP): the falsely predicted positive number. False Negative(FN): the falsely predicted negative number.

Then we can calculate the precision value and recall value. Precision is the ratio of correctly predicted positive data to the total predicted positive data. It shows the exactness of a classifier. Recall is the ratio of correctly predicted positive data to the actual positive data. It shows the completeness of a classifier. The F1 score can keep a balance of precision and recall. Thus, it can show both the exactness and completeness of a classifier. The formula (5), (6) and (7) shows the calculation for precision, recall and F1 score.

Precision = WX

WXYZX

Recall = WX WXYZ^

𝐹 =2 · cd#eF&Ff% ·ghe$ii

(5)

(6)

(7)

E cd#eF&Ff%Yd#e$ii

Especially for k-fold validation, there are various ways to calculate F1 score. The best way to is computing TP, FP and FN for each fold, and calculate the F1 score based on the “micro” metrics. That means we can sum the TP, FP and FN from each fold, and use the sum value to calculate F1. Because it can give us the most unbiased result [18].

Accuracy is the ratio of correctly predicted observation to the total observation. It is used to describe how accurate the model is. It can be calculated by the formula (8).

Accuracy = WXYW^ WXYZXYZ^YW^

(8)

Therefore, with accuracy and F1 score, I can not only evaluate the exactness and completeness of the classifier but also evaluate how accurate the classifier is.