Research on Two Feature Selection Techniques

Chang Liu
College of Engineering and Computer Science
Australian National University
u6726400@anu.edu.au

Abstract. Feature selection is a well-known data preprocessing technique. In today's world filled with massive amount of data with numerous data features, it is almost certain that some features in a dataset are redundant or irrelevant for meeting the research purposes. Hence, it is necessary to select features to improve the performance of research models as well as increase the time efficiency. This report examines two different feature selection technique – magnitude measures and genetic algorithms using VehicleX dataset to research their impact on the results. It is found that both techniques improve their model's performance and help reduce the computation time for training models.

Keywords: Feature Selection, Magnitude Measures, Generic Algorithms

1 Introduction

In real-world application, it is often believed that more data features mean more useful and detailed information for processing [3]. With the technology progressing nowadays, it is common for a dataset to have hundreds or thousands of data features, among which some features are surely redundant or irrelevant to a specific research purpose. Including those "bad" features will potentially decrease the research model's performance and add more computational complexity which requires more resources and time. That is when the feature selection techniques come in handy to address such issues.

In this report, two different feature selection techniques were introduced and implemented using VehicleX synthetic dataset. Section 2 will be a detailed description of the dataset used. Followed by that, the introduction of magnitude measures and its implementation in neural networks were covered in Section 3 and 4. Then, the Genetic Algorithms for feature selection was introduced. And the conclusion was drawn from the result discussion in Section 6.

This report aims to research the impact of feature selection techniques on the performance of the corresponding models. And it can be used in further research of feature selection related field study.

2 VehicleX Synthetic Dataset

VehicleX is a large-scale synthetic dataset generator that utilizes the graphics engine Unity to render synthetic vehicles and a Python API to generate detailed label data like car type and color [5].

The dataset used in this report consists of a total 1,362 vehicles annotated with detailed labels. The training, validation and testing data have 45,438, 14,936 and 15,142 data points respectively (shown in Table.1), of which each point has 2048 features extracted from ResNet and pretrained on ImageNet. The label data lists detailed labels including vehicle orientation, light intensity, light direction, camera distance, camera height, vehicle type, vehicle color and vehicle ID in the XML file.

Due to the low time efficiency to train the neural network on such massive amount of data with low-level machine, 2,000 data points are randomly chosen to be the training data, 500 points are chosen to be the test data and vehicle type (11 entries) is chosen to be the output target.

Table 1. Statistics of train, validation and test dataset

Split	Number of samples	Percentage of all
Train	45,438	60.17%
Validation	14,936	19.78%
Test	15,142	20.05%

3 Magnitude Measures of Contributions

To measure the significance of an input feature to the output, weight matrices of the network model is used. The research paper [1, 2, 4] described the following measure to calculate the contribution of an input to a neuron in the hidden layer:

$$P_{ij} = \frac{|w_{ij}|}{\sum_{p=1}^{ni}|w_{pj}|} \tag{1}$$

w_{ij} represents the weight from input feature i to a neuron j in the subsequent hidden layer. The denominator is the summation of the weights from all inputs to hidden neuron j. The measure is further extended to calculate the contribution of a hidden neuron to an output neuron as follows [1, 2, 4]:

$$P_{jk} = \frac{|w_{jk}|}{\sum_{r=1}^{nh}|w_{rk}|} \tag{2}$$

Similar to Equation (1), w_{ij} is the weight from a hidden neuron j to an output neuron k, and the denominator is the summation of all weights from hidden neurons to output neuron k. And the significance of an input neuron to an output neuron can then be measured through [1, 2]:

$$Q_{ik} = \sum_{r=1}^{nh}(P_{ir} \times P_{rk}) \tag{3}$$

Through the measures above using weight matrices of the neural network, the magnitude of contributions can be calculated, which represents how significant an input is to the output targets so that decisions can be made to remove unimportant input features or retain the important ones. Additionally, the technique can also be used in networks that have multiple hidden layers.

4 Neural Network Setup for Magnitude Measures

The neural network in this report is implemented using Pytorch. To increase the time efficiency of training, the customized neural network is set up to have 1 input layer with 2048 neurons, 1 hidden layer with 1500 neurons and 1 output layer with 11 neurons. A linear layer connects the input layer and the hidden layer, followed by a sigmoid activation function, and another linear layer connects the hidden layer and the output layer.

The parameters of the network are manually tuned with 50 iterations, batch size of 16 and learning rate of 0.01. In addition, the cross-entropy loss function and the Adam optimizer are used to train the network.

The baseline network model is trained using the 2000 randomly chosen data points with full 2048 features.

After applying the magnitude measures to assess the significance of each input feature to the outputs, a total of 100 out of 2048 features are selected as the most significant feature to retrain the network model. The selection of the 100 features is applied to the same 2000 training data points as well as the 500 testing data points, which are used to retrain the neural network model.

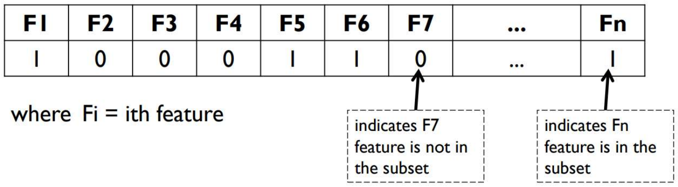
5 Genetic Algorithms for Feature Selection

The Genetic Algorithm (GA) is a heuristic search algorithm based on the concept of genetics and the theory of natural selection [3]. In nature, organisms' genes will evolve over generations to better adapt to the environment, which forms the inspiration of GA. It operates on a population of chromosomes to improve the quality of solutions and produce better approximations by utilizing three operators – selection, crossover and mutation.

On iteration in GA is similar to an evolutionary generation. The population (often in the form of bitstrings) is evaluated by fitness function, which could be maximized or minimized. After that, parents are selected based on their fitness, who are then used to generate the next generation of candidate solutions. The recombination of pairs of parents is conducted by the crossover operator, creating an offspring with split bits from both parents. The offspring may also undergo mutation, which is flipping bits in the creation of offspring candidate solutions.

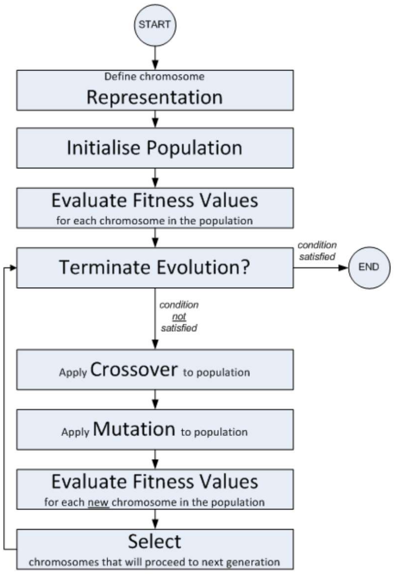
GA can be used in feature selection, where the selection status of each feature can be represented by a chromosome with binary string encoding. That is, the chromosome takes the form of a bitstring where bit "1" indicates the feature is selected whereas the bit "0" means not selected, as is illustrated in Fig.1.

Fig. 1. Illustration of feature representation in GA


F1
F2
F3
F4
F5
F6
F7
...
Fn
1
0
0
0
1
1
0
...
1
where Fi = ith feature
indicates F7 feature is not in the subset
indicates Fn feature is in the subset

The procedure flow of GA for feature selection is shown in Fig.2. Starting with defining the chromosome representation stated above, we initialize a population of chromosomes, which will be changed over iterations of GA. Then, we proceed with the evaluation of the fitness values assigned to each chromosome and step into the iteration of evolution where we apply crossover and mutation operators with certain probabilities, evaluate the new chromosomes and select the fittest ones for next generation.

Fig. 2. Process flow of feature selection using GA


START
Define chromosome Representation
Initialise Population
Evaluate Fitness Values for each chromosome in the population
Terminate Evolution?
condition satisfied
END
condition not satisfied
Apply Crossover to population
Apply Mutation to population
Evaluate Fitness Values for each new chromosome in the population
Select chromosomes that will proceed to next generation

In the GA part of this report, we used Support Vector Machines (SVM) to classify each data entry into 11 classes. Thus, the fitness function was set to be the accuracy score of the SVM model, and the goal is to maximize the accuracy of the classifier model. The parameters of GA used in this report are shown in Table.2.

Table 2. Parameters of GA

Encoding	Binary string

Size of population	20
Number of generations	10
Crossover operator	2-point crossover
Crossover probability	0.9
Mutation operator	Flip bit
Mutation probability	0.2
Selection	Tournament selection with tournament size of 2

6 Results and Discussion

The result of the baseline network model after 50 epochs is shown in Table.3, using all 2048 input features.

Table 3. Results of baseline model

	Loss	Accuracy
Train	415.5375	13.55%
Validation	442.7029	21.40%
Test	-	18.40%
Time spent	5min 39s	

Comparing with the result of the model with top 100 features selected by magnitude measures (shown in Table.4), we can see an increase in both training accuracy and testing accuracy, especially obvious with training accuracy, which indicates that using the neural network structure in this report, among all 2048 input features, there exist a certain number of features that are redundant or irrelevant negatively impacting the performance of the model. By selecting the top 100 features based on the magnitude measures, some of those “bad” features were removed and cannot further affect the model’s performance, which is why there is a rise in the model’s training and testing accuracy. Additionally, from the different times spent to finish the total 50 epochs of training, we can find that feature selection by the magnitude measures can substantially reduce the computation time when training neural network models.

Table 4. Results of feature selection using magnitude measures

	Loss	Accuracy
Train	217.4907	36.20%
Validation	268.2476	19.60%
Test	-	21.40%
Time spent	30s	

Due to the hardware limitation, the GA model was set to have population size of 20 with 10 generations instead of higher values which could potentially yield better results. The results of the best individual selected by GA are shown in Table.5.

Table 5. Results of the best individual in GA

Number of features selected	1041
Accuracy score	21.8%
Time spent	43min 13s

From the results, we can see that 1041 out of 2048 features were selected in the best individual after 10 generations. The accuracy score 21.8% is identical as the baseline SVM model with all features selected. Hence, we think the results of feature selection using GA is similar to the ones using magnitude measures. That is, there exists redundant features which do not positively contribute to the model’s performance.

7 Conclusion

In this report, two different feature selection methods were implemented – magnitude measures and Genetic Algorithms - using the VehicleX synthetic dataset. The magnitude measures utilize the weight matrices from input layer to its subsequent hidden layer to calculate the importance of each input feature. The GA represents the feature selection with binary encoding chromosome and evolves over generations to find the best individual with the optimal selection of features. Both methods prove that among all features of the dataset, there exist some redundant or irrelevant ones that negatively impact the performance of the model. And selecting significant features can also lower the computation time for training the model.

Although we drew the above conclusion based on the program results, it may not appear to be exactly accurate since we randomly chose only part of the dataset to proceed with the experiment due to the hardware limitation. For the future work, the whole dataset could be used in an advanced environment where the hardware can process such huge amount of data so that we can yield more reliable results to analyze and reach conclusions from. In addition, the GA can be set with higher parameter values and use other different classification models to generate better results.

Reference

1. Gedeon, T.D., 1997. Data mining of inputs: analysing magnitude and functional measures. *International Journal of Neural Systems*, *8*(02), pp.209-218.
2. Gedeon, T.D., 1996, November. Indicators of input contributions: analysing the weight matrix. In *1996 Australian New Zealand Conference on Intelligent Information Systems. Proceedings. ANZIIS 96* (pp. 166-169). IEEE.
3. Kannan, V., 2018. Feature selection using genetic algorithms. *Master's Projects.* 618. doi: https://doi.org/10.31979/etd.6mq4-cp5p
4. Wong, P.M., Gedeon, T.D. and Taggart, I.J., 1995. An improved technique in porosity prediction: a neural network approach. *IEEE Transactions on Geoscience and Remote Sensing*, *33*(4), pp.971-980.
5. Yao, Y., Zheng, L., Yang, X., Naphade, M. and Gedeon, T., 2019. Simulating content consistent vehicle datasets with attribute descent. *arXiv preprint arXiv:1912.08855*.
6. Zuo, S., An Implement of Network Reduction Technique Using Distinctiveness of Hidden Neurons.