Feature Selection Based on Sensitivity Analysis Method and Genetic Algorithm for Mark Prediction

Yue Ma[1],

[1] Yue Ma, Australian National University,
2601 Canberra, Australia
Yue.Ma@anu.edu.com

Abstract: In practice, the dataset often includes irrelevant and redundant features that affect model performance. Feature selection helps select those inputs which contribute most to the output. Input perturbation is one of the typical methods in the sensitivity analysis for selecting important features. The genetic algorithm is an advanced optimization method that can be used to select features. In this paper, we implement input perturbation and genetic algorithm (GA) to select essential features. The final testing accuracy of the artificial neural network(ANN) is better than that of the dataset paper, reaching 80%.

Keywords: Mark Prediction, Feed-forward Neural Network, Feature Selection, Sentiment Analysis, Genetic Algorithm

1 Introduction

Mark prediction based on partial marks is useful in educational fields. The final mark prediction can motivate students who are not satisfied with the result to work hard to prepare the final exam. Educational institutions can use the prediction model as a tool to make decisions as well.

The dataset sources from an undergraduate computer science subject COMP 1111 at the University of New South Wales, consisting of 153 observations' information in this subject [1]. The assessments include labs, assignments, and mid-term quiz accounting for 40% of the total marks and the rest 60% being the final exam. The final marks consist of four categories: Distinction or above, Credit, Pass, Fail. The project problem is to predict students' final marks of COMP 1111 based on 40% partial marks, which is also the dataset paper problem.

Irrelevant and redundant features do not contribute or sometimes reduce model performance. It is recognized that using significant inputs can push the neural network to its limits of capability and give us the best possible results [2]. The feature selection is the process of selecting a subset of relevant features for model construction. The technique paper discussed several techniques to measure the significance of the inputs, including magnitude techniques, sensitivity analysis, and functional measures. Sensitivity analysis is used to measure the effect an input has on the output of the network [3]. The larger the effect is, the more significant the input is. The technique paper discussed two methods of the sensitivity analysis: the partial derivative method and the input perturbation method. We choose the second method to select features. The drawback of this technique is computationally expensive, that exhaustive combinations of features become impracticable when the number of features is big.

One of the most advanced algorithms for feature selection is the genetic algorithm, which has been widely used as a search algorithm for a wide range of optimization problems [4]. GA is inspired by the law of natural evolution. The individuals tend to evolve over generations to better adapt to the environment. The search evolves a population of chromosomes using crossover, mutation, and selection methods [5]. At each generation, individuals are selected according to the fitness value. Then the new population is created from these individuals by the process of crossover. The offspring might also undergo mutation.

In this paper, we first evaluate the input perturbation method with an artificial neural network (ANN), then apply the genetic algorithm to select important features.

2 Data Preprocessing

Each assessment in COMP 1111 dataset has missing values. We used the median to impute the missing values since extreme value drags the mean to its side. The aggregate final marks are categorized into four classes [6]. The other 10 inputs are normalized with the mean of 0 and a standard deviation of 1. Table 1 lists the first five records.

Table 1. The preprocessing dataset.

No.	lab2	tutass	lab4	…	f1	mid	lab10	final
0	0.807539	0.050926	0.225924	…	1.011863	1.374396	0.349504	2.0
1	0.908482	1.433211	0.638236	…	1.011863	0.779425	0.068263	2.0
2	0.908482	0.050926	1.090084	…	1.011863	0.955907	0.068263	4.0
3	0.807539	0.050926	0.638236	…	1.011863	0.184454	0.349504	3.0
4	0.807539	0.320108	1.502396	…	1.011863	0.608841	0.349504	3.0

3 Network Topology and Performance Measurement

In this project, we modeled a 10-5-4 artificial neural network. In assignment 1, we tried to experiment on a three-layer neural network, which always caused overfitting. The dataset is so small that we chose a two-layer network, which has the same testing accuracy as previous. We chose the ReLU function as hidden neurons' activation and the SoftMax function as output neurons' activation function. The cross-entropy loss is used to calculate the error of the model. For reliable results, we used 10-folds cross-validation to get the averaged loss, accuracy for ten runs to evaluate the model performance.

4 Techniques

4.1 Sensitivity Analysis

The idea of sensitivity analysis is that by slightly varying each input and comparing the results of the output, the one leads to the large change is the significant input. The technique paper discussed two neural networks-based sensitivity analysis methods: the partial derivative and the input perturbation. The partial derivative method identifies the significant input by taking the partial derivative of the output of the neural network concerning the input. The Jacobian matrix calculates the partial derivatives of output concerning inputs [6].

The input perturbation method implements a small perturbation on each input of the network and measures changes in the outputs. Each time, one input increase or decrease a perturbation while other inputs do not change. Then the input that has the largest effect on the outputs is considered the most significant [7]. In this project, we used the mean square error (MSE) to calculate changes in the output of the neural network after perturbation. To select features, we took experiments on ±1, ±2, ±3 input perturbation with each input keeping other inputs fixed. The rank of inputs is listed in Table 2. The inputs' contribution to each output was plotted in Fig 1. The drawback of this method is computationally expensive.

4.2 Genetic Algorithm and Feature Selection

The genetic algorithm is a heuristic optimization method inspired by the theory of natural evolution. The main driving operators of a genetic algorithm (GA) is the selection (to model survival of the fittest), recombination through the application of a crossover operator (to model reproduction), and the mutation [8]. At each generation, offspring are made up of selected better quality chromosomes then become the population in the next generation. Crossover and mutation operations are applied to chromosomes to achieve diversity in the population and reduce the risk of the search being stuck with a local optimal population [5].

The goal of this project is to use GA to select useful features for an ANN. In this project, each chromosome was represented as a binary string, encoded 10 features as its components. Genes have binary values: 0 or 1, which represents selecting or reducing a particular feature.

The initial population was randomly initialized by Numpy, then the parents are selected from the population. The selecting criterion is the fitness value related to each chromosome. The fitness function returns testing accuracy for each network. Offspring are created by crossover and mutation operators.

The crossover operator exchanges genes between parents. In this project, we use the one-point crossover which randomly selects a cross-over point and swaps the bitstrings after the point between two parents.

The mutation operator is applied to introduce the new genes in the offspring. We use the random mutation that randomly chooses bits to flip.

The stop criterion is greater than the number of generations. Fig 1 shows the process of the genetic algorithm for final mark prediction.

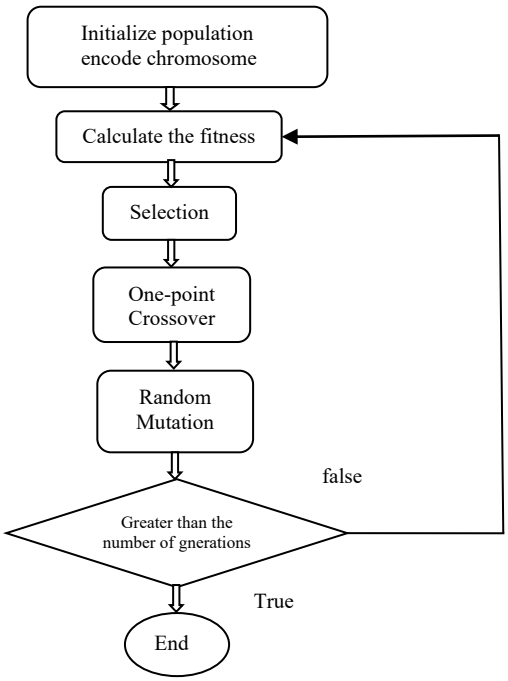

Fig. 1 The architecture for a GA+ANN model for final mark prediction.

5 Results and Comparison

5.1 Input Perturbation Technique

We took input perturbation and calculated the changes of output by mean square error MSE. The contribution of inputs to each of the 5 outputs was averaged to determine the significance of inputs to the network. Table 2 shows the results of six experiments. In the most and less significant inputs, results are quite stable. “mid” is always the most important inputs with more than twice MSE value as the other inputs. The top three significant features are “mid’, “h1” and “p1” while the bottom three significant features are “lab7”, “tutass” and “lab10”.

Table 2. The significance of each input.

Perturbation	Most Significant								Less Significant	
+1	mid	h1	p1	lab4	f1	lab2	lab7	h2	tutass	lab10
-1	mid	h1	p1	lab4	lab2	f1	h2	lab7	lab10	tutass
+2	mid	h1	p1	f1	lab4	h2	lab2	lab7	tutass	lab10
-2	mid	h1	p1	lab4	lab2	h2	f1	lab7	lab10	tutass
+3	mid	h1	p1	f1	lab4	h2	lab2	lab7	tutass	lab10
-3	mid	h1	p1	lab4	lab2	f1	h2	lab7	lab10	tutass

Fig 2 shows the inputs’ contributions to each output when inputs are perturbed with ±1. The significant inputs are relatively consistent in two models.

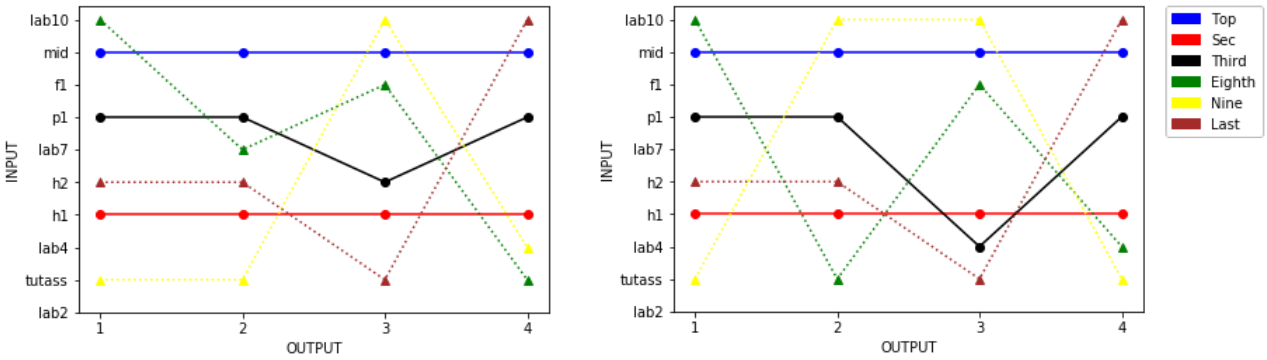

Fig. 2 Contributions of top three significant inputs and bottom three significant inputs to each output.

We modeled an artificial neural network with a 10-5-4 structure. The base represents the performance of the network fed with all inputs. We then reduced significant inputs or less significant inputs. Based on the result above, we feed the top five important features or bottom five features into the network. The plot below indicates that the overall number of correct of the network with top five significant features was not reduced. But when the top five significant inputs were removed, the overall performance was reduced significantly.

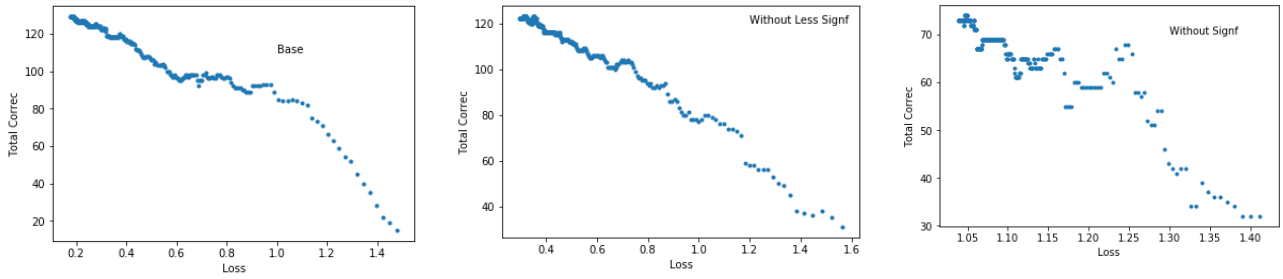

Fig.3 Total correct vs. Loss

Table 3 shows the averaged training accuracy and testing accuracy for the two models. The performance is evaluated by averaged accuracy of 10-fold cross-validation. Compared with the base testing accuracy which is 63.49%, the testing accuracy of the network with significant features reaches 74.57%.

Table 3. Compare the performance of the networks.

	ANN		ANN (Without Less Signif)	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
Run 1	86.88%	58.20%	81.02%	68.22%
Run 2	84.78%	44.11%	79.44%	68.19%
Run 3	83.71%	58.78%	79.17%	72.84%
Run 4	82.05%	79.72%	77.52%	90.98%
Run 5	81.30%	71;48%	76.73%	86.38%
Run 6	84.24%	74.88%	79.37%	86.30%
Run 7	79.35%	59.15%	82.22%	67.48%
Run 8	79.56%	72.67%	79.48%	73.33%
Run 9	85.70%	53.62%	80.07%	66.06%
Run 10	83.28%	62.30%	81.32%	65.88%
Ave	83.09%	63.49%	79.63%	74.57%

5.2 Feature Selection by Genetic Algorithm

We change several parameters including the number of generations, population, mutation rate. The final selected significant inputs and the models' testing accuracy are listed in Table 4.

Table 4. Significance inputs

Model	Selected Significant Inputs					ANN Testing Accuracy
1	2	5	8	9	-	76.67%
2	2	4	5	8	9	76.67%
3	2	3	5	8	9	80%
4	5	9	10	-	-	76.67%
5	3	5	9	10	-	80%
6	2	5	9	10	-	76.67%
7	2	6	8	9	10	76.67%
8	3	5	8	9	10	76.67%
9	2	4	7	8	9	80%
10	2	5	6	8	9	73.3%

In Table 5, compared with the TG testing accuracy (66%) in the dataset paper, the testing accuracy of the network with significant features selected by the input perturbation technique is better, at 70.08%. The "ANN+GA" model is the best with the highest testing accuracy, reaching 80%.

Table 5. Compare the performance of the networks.

Model	Ave. Test Accuracy
ANN (dataset)	53.8% (Ave. A-J) 66.0% (Run TG)
ANN (base)	63.49%
ANN+IP	74.57%
ANN+GA	80%

Fig 4 shows that the genetic algorithm converges. The final testing accuracy ranges from 73.3% to 80%.

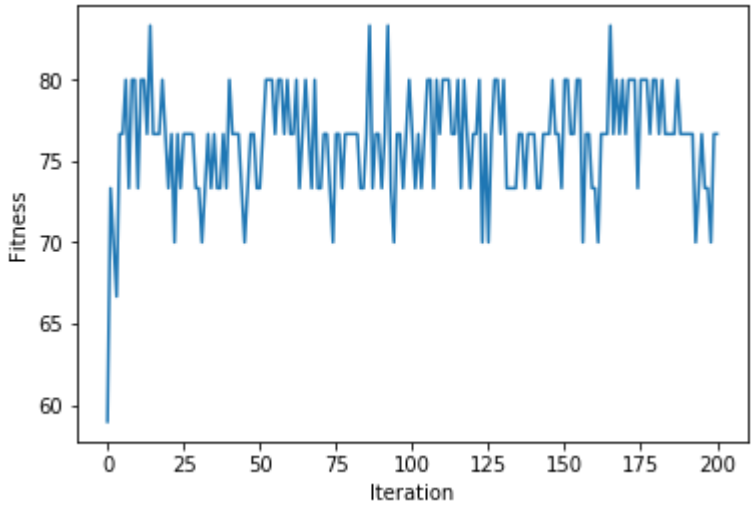

Fig.4 Fitness vs. iteration

6 Conclusion and Future Work

We first trained a 10-5-4 feed-forward neural network and chose the ReLU as the activation of the hidden neurons, using back-propagation and cross-entropy loss function. We then used the input perturbation technique to rank the features' importance. The top five and bottom five features were feed into the ANNs. The result shows that the model without significant features reduced, and the model accuracy with significant features increased to 74%. Finally, we use the "ANN+GA" model to select features. The final testing accuracy reaches 80%.

In this project, we only used the genetic algorithm to select important features, however, when the input size changes, the structure of the network would better to change. Therefore, the genetic algorithm is considered to optimize the network structure in the future.

References

1. Choi, Edwin Che Yiu, and Tamis Domonkos Gedeon. "Comparison of extracted rules from multiple networks." In Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, pp. 1812-1815. IEEE, 1995
2. Milne, Linda. "Feature selection using neural networks with contribution measures." In AI-CONFERENCE-, pp. 571-571. WORLD SCIENTIFIC PUBLISHING, 1995
3. Gedeon, Tamás D. "Data mining of inputs: analysing magnitude and functional measures." International Journal of Neural Systems 8, no. 02 (1997): 209-218
4. X. Niu, L. Chen and Q. Chen, "Research on genetic algorithm based on emotion recognition using physiological signals," in 2011, . DOI: 10.1109/ICCPS.2011.6092256.
5. N. Sharma and T. (. Gedeon, "Hybrid genetic algorithms for stress recognition in reading," in 2013, . DOI: 10.1007/978-3-642-37189-9_11.
6. Gedeon, T. D., and S. Turner. "Explaining student grades predicted by a neural network." In Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), vol. 1, pp. 609-612. IEEE, 1993
7. Dimopoulos, Yannis, Paul Bourret, and Sovan Lek. "Use of some sensitivity criteria for choosing networks with good generalization ability." Neural Processing Letters 2, no. 6 (1995): 1-4
8. Engelbrecht, Andries P. Computational intelligence: an introduction. John Wiley & Sons, 2007