What is Good Feature? Feature Selection for Deep Learning

Yunsung Chung
College of Engineering & Computer Science
Australian National University
Acton ACT 2612 Australia
u6934323@anu.edu.au

Abstract. As the neural network model's performance advances are clearly limited, the importance of data is increasing. When a lot of data is used to derive a result from the inside, too much data degrades the model's performance, and sometimes the desired derived value cannot be obtained. Since it is rare to create a model with well-refined data in an actual field, selecting proper features is a part that can greatly contribute to the improvement of model performance. Therefore, various methods such as Garson's algorithm, Gedeon's algorithm, etc. for selecting features have been previously introduced. However, these methods are limitedly tested on a simple neural network. This paper experiments the data analysis methods on multilayer perceptron to check whether methods are acceptable in deep learning. The experiments shows that the methods are effective for feature selection even in deep learning environment.

Keywords: Data analysis, data pruning, magnitude analysis, functional analysis, pupillary response, neural network, deep learning, anger dataset, multilayer perceptron, hidden neuron

1 Introduction

The model trend these days is more and more layers and more complex configurations. Therefore, we try to prove whether the algorithms such as Gedeon's algorithm, Garson's algorithm, etc. in [1] used for feature selection in a simple neural network are indeed effective in deep learning models. The network used is a multilayer perceptron (MLP) method, and training is performed through the backpropagation method. All networks are fully connected and consist of eights layers, and each layer except for the output layer is activated by the rectified linear activation (ReLU). All neurons are input as values of the next layer through weights and bias values, and the weight is corrected using the backpropagation technique of the final output. For the loss function, the loss value was obtained through the Mean Square Error method, and the Total Sum of Squares (tss) is used in a specific analysis method as needed. Train data and test data are completely separated so that test data cannot be accessed during training, and test data is configured so that there is no overlapping value with training data. It should be noted that this is a network constructed for the purpose of finding correlation between data, not an experiment to increase model performance and model accuracy. Finally, the logistic sigmoid function was used as the activation function.

2 Method

The data analysis method was classified into two types as suggested in [1] and proceeded. Firstly, magnitude analysis was performed, and secondly, functional analysis was performed. Each analysis has a technique that is an evaluation criterion, and performance is evaluated based on it.

The first part, magnitude analysis, compared three techniques which are Garson's approach, Milne's approach and Gedeon's approach based on the Brute force technique. To measure which input is significant, we used the weight matrix of the trained network, and measure it in different ways.

The second part, functional analysis, compares four techniques. This is a technique that measures the functional contribution of the input to measure which input has critical information for predicting the output. Two methods measured using the weight matrix of the input and trained network, and each input and weight matrix. It uses two augmented methods.

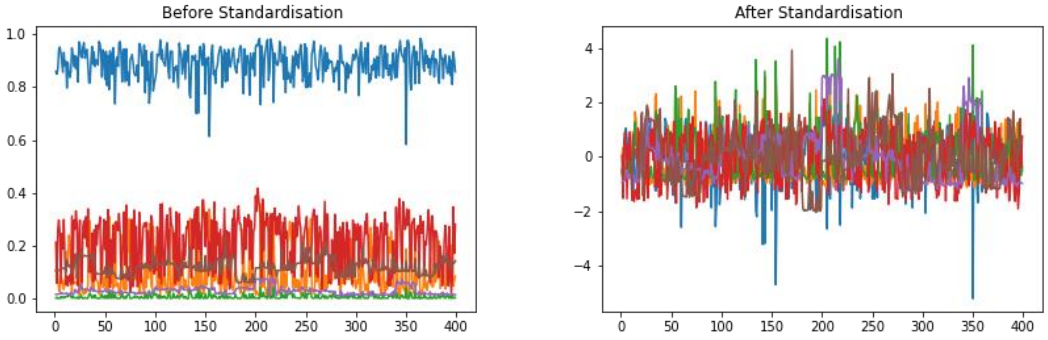
2.1 Dataset – pupillary response of anger

The dataset is data that measures whether the angered emotion is the real anger, or the acting anger based on the pupil response data. The data has a total of 8 features: Index, Video, Std, Mean, Diff1, Diff2, PCAd1, PCAd2. Details for each input are as follows:

* Mean: The mean of in pupillary response
* Std: The standard deviation of in pupillary responses

* Diff1: The change of left pupillary size after watching a video
* Diff2: The change of right pupillary size after watching a video
* PCAd1- An orthogonal linear transformation with first principal component
* PCAd2- An orthogonal linear transformation with second principal component
* Label – The Genuine or Posed emotion.

Among them, six features were used for training, excluding Index and Video, which are unnecessary features for training. In addition, each input consists of 400 vectors, and the train data, evaluation data, and test data were split at 6 to 2 to 2 ratios. That is, the training data has 240 samples and the evaluation and test data have 80 samples. For the output, Genuine and posed are replaced respectively with 1 and 0.



Standardisation was carried out to maximise fairness because of the range of each feature is scattered. After calculating the mean (μ) and the standard deviation (σ) for each feature, Standardisation was performed using the follow equation:

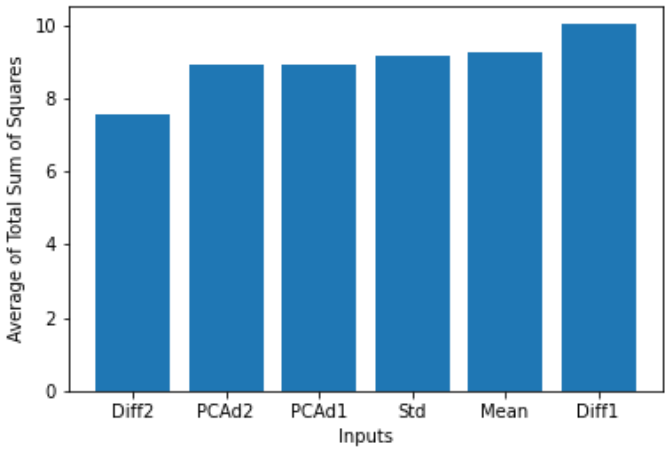
$$z = \frac{x - \mu}{\sigma}$$

2.2 Network configuration

The model consists of a 6-6-12-24-48-24-12-6-1 topology consisting of 6 inputs,126 hidden neurons, and 1 output. Training was performed with learning rate = 0.1 and epochs = 100,000, the mean squared error was used as the loss function, and the stochastic gradient descent was used as the optimizer.

2.3 Brute force technique

The method suggested in [1] was used as it is. In order to maintain a consistent line, 2 inputs out of 6 inputs were excluded. This could constitute 15 combinations. After training the model with the remaining inputs, the total sum of squares was calculated as test data. Each network was run 10000 times, and before training for each input, all variables of the network were initialized to prevent redundant learning.



The result shows the average of total sum of squares values by inputs in increasing order. The average value of the total sum of squares value was calculated as the average value of 10 combinations in which each input was not excluded among 15 combinations. The results were more stable than the results presented in [1], which means that the data used in the experiment does not have a significant difference in the importance of each feature. This result is a standard indicator for the magnitude analysis technique to be followed and a measure of the performance of each technique.

2.4 Magnitude analysis techniques

As for the technique used for comparison, three techniques which are Garson's approach, Milne's approach and Gedeon's approach suggested in [1], [3], and [4] were implemented. The Garson's algorithm introduced in [3] as follows.. This algorithm measures the significance of each feature through weight of neurons from the input to the next layer. It has the disadvantage that positive and negative weights are cancelled and some contributions are lost.

$$G_{ik} = \frac{\sum_{j=1}^{nh} \frac{W_{ij}}{\sum_{p=1}^{ni} W_{pj}} W_{jk}}{\sum_{q=1}^{ni} \left(\sum_{j=1}^{nh} \frac{W_{qj}}{\sum_{p=1}^{ni} W_{pj}} \cdot W_{qj} \right)}, where \ \mathrm{W_{ij}}: \text{weight between i-th and j-th layer}$$

The below equation is a formula introduced in [4]. The difference of Milne's approach from Garson's approach is used absolute values in denominators to compensates for shortcomings in [3] and has the advantage of having a clear sign, but still has the disadvantage that the denominator does not have a clear meaning.

$$M_{ik} = \frac{\sum_{j=1}^{nh} \frac{W_{ij}}{\sum_{p=1}^{ni} |W_{pj}|} \cdot W_{jk}}{\sum_{q=1}^{ni} \left(\sum_{j=1}^{nh} \left| \frac{W_{qj}}{\sum_{p=1}^{ni} W_{pj}} \cdot W_{qj} \right| \right)}, where \ \mathrm{W_{ij}}: \text{weight between i-th and j-th layer}$$

Finally, the magnitude contribution of the input can be measured to the output by Gedeon's algorithm presented in [1]. In addition, it has a clear sign, which is an advantage of [3], as an advantage.

$$P_{ij} = \frac{|W_{ij}|}{\sum_{p=1}^{ni} |W_{pj}|}, \text{where } \mathrm{W_{ij}}: \text{weight between i-th and j-th layer}$$

The above equation calculates proportions of weight from input to hidden neurons. All numerator and denominator are calculated as absolute values to measure only the ration regardless of positive or negative weights.

$$P_{jk} = \frac{|W_{jk}|}{\sum_{p=1}^{ni} |W_{rk}|}, \text{where } \mathrm{W_{jk}}: \text{weight between j-th and k-th layer}$$

In order to consider significant of each feature from the hidden layer to the output layer, [1] introduced an equation above to go beyond the method that considers only the weight of each input feature. $\mathrm{P_{jk}}$ is an equation that calculates the ratio of hidden neuron weight to output.

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

Therefore, the above equation shows the importance of each feature through input to output without the problem of sign in the contribution.

2.6 Functional analysis techniques

As for the technique, the technique used in [1] and [5] and the aggregated technique were used for comparison. The angle between each vector is calculated by the following equation.

$$angle(i,j) = \tan^{-1}\left(\sqrt{\frac{\sum_{p}^{pats} sact(p,i)^2 * \sum_{p}^{pats} sact(p,j)^2}{\sum_{p}^{pats} (sact(p,i) * sact(p,j))^2}} - 1 \right)$$

In this equation, the measurement technique is classified according to the equation that composes sact. The method suggested in [5] measures the functional differences based on the hidden weights constituting the hidden layer in the input, and the formula to construct them is as follows:

$$where \ sact(p,h) = norm(weight(h)) - 0.5$$

In [1], functional differences are measured based on the pattern of the input. So the formula to construct this is:

$$where \ sact(p,h) = pattern(h) - 0.5$$

i and j are indices for arbitrary inputs and are a method of measuring the correlation by measuring the angle of the vector of each input. Through this, the functional difference can be confirmed, and the higher the degree, the lower the correlation between each vector, which means that it possesses critical information. Therefore, if the aggregated form model is used and the average angle with other input vectors is calculated, it will be possible to measure the importance more clearly.

3 Results and Discussion

In the following table, results of the magnitude analysis for each model are listed in the order of the most significant to the least significant. The results are based on the Brute force technique which is Model B, and each model will divide the 6 measured features into the top 3 and the bottom 3 and compare the proportions consistent with it. In order, model Q represents Gedeon's algorithm, Model G represents Garson's algorithm, and Model M represents Milne's algorithm.

	Most significant					Least significant
Model Q	PCAd1	Std	PCAd2	Mean	Diff2	Diff1
Model G	PCAd1	Std	Diff1	Diff2	Mean	PCAd2
Model M	Diff1	Diff2	PCAd2	Mean	Std	PCAd1
Model B	Diff2	PCAd2	PCAd1	Std	Mean	Diff1

As a result, all three have shown tremendous advantages in terms of speed. In the case of Model B, since all combinations must be trained, the computation time increases exponentially depending on the number of features in dataset and the complexity of the model. However, the remaining three algorithms have a temporal gain because they can be calculated using only the weight of the trained neurons after one time training. Model Q and Model M show 66.6%, and Model G shows 33.3% in accordance with Model B.
The table below shows the results of the functional analysis. Model W is a model showing the functional differences of hidden weights, and Model U is an aggregated form of Model W. Likewise, Model I is a model that calculates the functional differences of the input pattern, and Model C is in its aggregated form. Models U and C measure significance through the average of the angles of the remaining inputs for a specific input, while model W and model I set thresholds to count cases that fail to exceed them and set the importance in the order of the lowest count.

	Most significant					Least significant
Model W	Diff1	Diff2	PCAd1	PCAd2	Mean	Std
Model U	Mean	PCAd2	PCAd1	Std	Diff1	Diff2
Model I	Mean	Diff1	PCAd1	PCAd2	Diff2	Std
Model C	Diff1	Std	Diff2	PCAd2	Mean	PCAd1

As a result, Model U and C shows 33.3%, and Model U and I shows 0% in accordance with Model B. There are two possible reasons for this problem. First, it is the difference in complexity of the model. Compared with the complexity of the model used in [1], the model used in this experiment is insufficient to judge the influence on the output based on the input pattern and the weights of the first hidden neurons because much more hidden neurons were used. The second is expected to be related to the diversity of features of the data. As for the data tested in [1], if you look at the results of the Brute force technique, you can see several features with a large difference between the general features in tss values. However, since the data used in this experiment consisted of features based on the difference in the calculation method based on the pupil size, the correlation between the features is quite high. Therefore, it cannot be seen that the angle calculated in the functional analysis represents a meaningful result between 0 and 1.

4 Limitation

The experiment used the same topology with limited hyperparameters, but there are several limitations. First, dataset used in the experiment is limited. If we could test the above-introduced analysis method on various data with various features and various distributions, we would have produced more reliable results. The data used in the experiment lacks diversity between features, unlike the data used in [1] because features only depend on how to calculate pupil size. In other words, classification is possible with only one feature of the dataset. In addition, a classifier such as the Adaboost classifier exhibits better training results and performance due to the nature of the data, but the weight value of the hidden layer constituting the deep network is required for calculation, indicating that the deep network was configured for the measurement value.

Although the Brute force technique, which is the criteria of comparison, was performed with the combination of pairs of inputs suggested, the result was unstable. Due to the nature of training, the possibility of falling into a local minimum

cannot be ruled out, so the evidence may seem insufficient to be a clear criterion. Each time of training can cause a slight difference in results.

5 Conclusion and Future Work

In summary, the feature contribution analysis methods used in neural network were proved to be suitable methods that can be used in deep learning. Despite the use of a relatively small dataset, Gedeon's algorithm not only shows a tremendous advantage in terms of speed compared to the other algorithms like the Brute force technique, but also shows the high accuracy in the result. Therefore, this proved to be very effective in pruning features of complex datasets in deep learning.

It was confirmed that the previously introduced method has limitations when analysing datasets with little difference in characteristics between features. Therefore, it is expected that research on functional analysis techniques that can overcome these limitations will be conducted. In addition, research can be conducted on whether the introduced methods can be applied to models such as BERT and GPT-3, which are more complex.

References

[1] Gedeon, Tom. (1997). Data Mining of Inputs: Analysing Magnitude and Functional Measures. Int. J. Neural Syst.. 8. 209-218. 10.1142/S0129065797000227.
[2] Chen, lu & Gedeon, Tom & Hossain, Md & Caldwell, Sabrina. (2017). Are you really angry?: detecting emotion veracity as a proposed tool for interaction. 412-416. 10.1145/3152771.3156147.
[3] Garson, G.D.. (1991). Interpreting neural network connection weights. AI Experts. 6. 47-51.
[4] Milne, Linda. (1999). Feature Selection Using Neural Networks with Contribution Measures.
[5] Gedeon, Tom. (1995). Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour. 26-29. 10.1109/ANNES.1995.499431.