Feature Extraction Effect of Autoencoder and Effect of Input Pruning on Small Dataset: The 'Fight or Flight' Response Case

Shiyu Chen

Research School of Computer Science,
Australian National University
{u5959123}@anu.edu.au

Abstract. *Objectives:* This paper aims to classify deceive responses by using a single question's response. *Approach:* A neuron network model is built as a baseline model. Feature extraction method such as autoencoder is used to improve model's identification power. This paper stacks the optimal autoencoder model with the baseline model together to classifies the deceive response. The input pruning method is the other method to enhance model performance. This paper performs input pruning by detecting distinct features. Finally, this paper combines the autoencoder with input pruning to verify the effect of overlaying two methods. *Main Result:* The classification accuracy of the baseline model is ambiguous. But the employment of autoencoder and input pruning enhances the model classification capability. And the combination of the two methods further improves identification power. ROI connections whose indices are 4, 12, 13 and 17 are the distinct features identified by distinctiveness analysis.

Keywords: Input pruning, functional measures, autoencoders, neuron network.

1 Introduction and Task Selection

This paper reproduces and extends the *network physiology of 'fight or flight' response in facial superficial blood vessels* (FOF) produced by Derakhshan, et al. (2018). FOF analysis the 'fight or flight' response in the psychology aspect while this paper focuses on the technical aspect.

1.1 Introduction of *Network physiology of 'fight or flight' response in facial superficial blood vessels*

FOF aims to identify falsehood responses of questions through the changes in blood flow in the cutaneous superficial blood vessel. Perinasal, chin, cheeks, periorbital and forehead are regions of interest (ROIs) selected by researching literature as potential indicators of lying response.

The experiment separated participants into deceiving and truthful groups. The deceive group participants were asked to 'steal' a gold necklace in the experiment room and lie when queried. The honest group was not instructed to 'stole'. Then, an interviewer asked participants eight yes or no questions. Questions are set based on the comparison question test (CQT) principle, categorized as relevant, irrelevant, and neutral questions. Thermal changes of each ROI were recorded when participants were answering each question. Twenty connections were created by pairing ROIs and it works as potential indicators of 'deceive or truth' response. FOF employed decision tree classifier, K-nearest neighborhood, linear discriminant analysis and support vector machine methods to excavate the relationship between connectivity indicators and 'deceive and truth' response. FOF selected the four most discriminative features by evaluating the absolute value of paired sample t-test, relative entropy, receiver operating characteristic and two-sample unpaired Wilcoxon test.

FOF result shows that the 'deceive or truth' response cannot be classified when all connections are included in the classification model as the average accuracy rate is 56.34%. But the classification is more successful when it solely takes the four most discriminative features (forehead-periorbital, chin-forehead, chin-perinasal and periorbital-forehead) into the model. The accuracy rate rises to 78.31% on average. It indicates that the connectivity of forehead-periorbital, chin-forehead, chin-perinasal, and periorbital-forehead is meaningful in identifying 'deceive or truth' response.

1.2 Task Selection

This paper aims to find a baseline neuron network model to predict the 'deceive or truth' response and enhance the model's prediction power by using autoencoder and input pruning.

This paper analyzes the relationship between twenty potential ROI connections and 'deceive or truth' response. FOF builds a machine learning classification model with reactions to the CQT question set. But the dataset provides information of question 6 only, which is a basic straight question asking if the tester stole the necklace. So, this paper analyzes the relationship between potential ROI connections and a single straightforward question's response. There is insufficient research nor reliable research result to support the validity of CQT and the classification result of CQT may be distorted due to multiple flaws and biases [2]. Thus, it is possible and worthwhile building a neuron network model to test the classification capability regarding a single direct question.

In addition, FOF's result indicates that the 'deceive or truth' classification performance of traditional machine learning models (i.e., decision tree (DT), support vector machine (SVM)… etc.) are ambiguous. But the same models realize satisfier result (i.e., 78.31% accuracy on average) when performing feature selection on the dataset. However, feature selection on the dataset may not be suitable for all cases. No features may be discriminative after completing feature selection for some cases. So, the second object of this paper is to optimize classification accuracy on the model side by using autoencoder.

Finally, this paper performs input pruning by data mining the input contribution to the neuron network. The feature selection methods in FOF are pure statistics, which focus on analyzing the pattern values. However, the features are distinct in pattern level do not mean it contributes significantly to the underlying machine learning model nor high classification capability. Thus, this paper analyzes the contribution of input features to the model by functional measures, prune out indistinct input features and evaluate the classification performance using distinct features. Then, the distinct features selected by input contribution are supposed to be compared with the feature selected in FOF. However, the provided dataset does not have labels. Therefore, it is impossible to compare the features generated in this report and FOF.

2 Data Inspection & Data Preparation

Data may need to be modified to fit the model and the outliers should be identified and be removed before building model. Thus, this paper examines data first.

2.1 Data Inspection

The dataset contains 31 patterns. Each pattern has 20 features and one binary output. Each feature is a continuous number, and all features have been normalized into zero and one. The dataset is equally distributed with 15 patterns in deceive group and 16 patterns in the truth group. The details of the dataset description are presented in Table 1 and Figure 1.

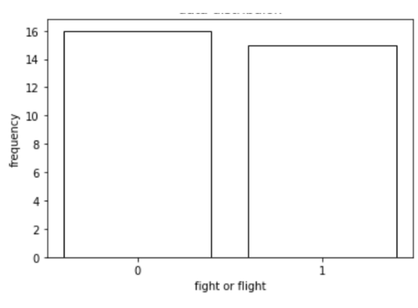
The most significant issue of this dataset is the number of data is small. Varies of techniques can be employed in this dataset to improve the performance. Firstly, shallow neuron networks are supposed to be employed because deep learning may cause overfitting when training small data sets. Each pattern is valuable in the small dataset. Cross-validation can be used to avoid wasting data in the test set.

The other issue is that the dataset is sorted. The first 15 patterns are in deceive group and the last 16 patterns are in the truth group. The Neuron network may be significantly biased because it spends the first half of time learning the characteristic of the deceive group and spend the other half of time distorts the learned weight and force the neuron network bias to the characteristics of the truth group.

Table 1. Summary statistics of dataset

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
count	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00
mean	0.47	0.33	0.43	0.45	0.41	0.40	0.42	0.36	0.39	0.39	0.34	0.36	0.39	0.38	0.33	0.37	0.40	0.38	0.42	0.29	0.48
std	0.31	0.23	0.30	0.25	0.30	0.33	0.27	0.22	0.26	0.25	0.21	0.25	0.29	0.28	0.27	0.27	0.33	0.30	0.27	0.22	0.51
min	0.04	0.00	0.04	0.09	0.03	0.00	0.11	0.03	0.06	0.12	0.04	0.01	0.02	0.04	0.01	0.02	0.01	0.02	0.04	0.03	0.00
25%	0.19	0.16	0.21	0.24	0.18	0.17	0.19	0.21	0.19	0.20	0.15	0.18	0.19	0.21	0.12	0.17	0.16	0.18	0.19	0.09	0.00
50%	0.39	0.27	0.39	0.38	0.37	0.27	0.33	0.28	0.33	0.31	0.33	0.27	0.32	0.27	0.25	0.33	0.30	0.26	0.39	0.25	0.00
75%	0.70	0.46	0.61	0.63	0.59	0.63	0.57	0.49	0.52	0.51	0.46	0.46	0.59	0.54	0.50	0.53	0.62	0.55	0.63	0.45	1.00
max	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.87	0.98	1.00	0.87	1.00	1.00	1.00	1.00	0.89	1.00	1.00	0.88	0.80	1.00

Figure 1. Distribution of Dataset Output



2.2 Data Preparation

This paper shuffles the dataset in order to avoid overfitting one class when training. In addition, leave one out cross-validation is used to calculate the accuracy to make the most use of the dataset.

3 Methodology

This report builds a baseline neuron network model first and then enhance the model by using autoencoder and input pruning.

3.1 Neuron Network Classification Model

Firstly, this paper builds a neuron network model to test the classification capability of 20 features and 'deceive or truth' response on the straightforward question. Studies have researched that the neuron network model performs better than DT, random forest, SVM and linear regression (LR) on average [1, 3] and on small dataset [10]. Furthermore, research also finds that the shallow neuron network work model performs better than the deep neuron network model [1]. So, neuron network is an optimal machine learning algorithm in building this model. In addition, DT, SVM and LR algorithm perform weakly in FOF's baseline case. But FOF has not tried neuron network. Therefore, a Neuron network model may provide unexpected classification accuracy. So, this report uses a shallow machine learning model to classify output from input features.

Learning rate, running epoch, hidden neuron and layer number are the hyperparameters of the neuron network model. This paper aims to build a shallow neuron network model, so the model should have no more than two hidden layers.

Paper starts modelling by tuning one-hidden layer network (two-layer network model). The sigmoid function is differentiable at all points, which is computationally efficient for two-layer networks. Thus, this two-layer network model uses the sigmoid function as an activation function. Root mean square propagation (RMSProp) is faster than stochastic gradient descent while maintaining the most property in the stochastic. Thus, RMSProp is employed when performing gradient descent. The model uses the cross-entropy loss function as it is the classic loss function for the binary output model. A two-layer network with four hidden layer neurons is set as the initial point. Then, this paper adds on neurons after the optimal hyperparameters of the current model are identified. The learning rate is turned from large to small. The final learning rate should be set at the point where the loss gradually decreases and then reaches a platform as the number of training epoch increases. The number of training epoch is also turned from large to small. Finally, the combination of learning rate and training number with the highest classification accuracy is the optimal hyperparameter set for the current model.

For the three-or-more layer neuron network, randomized leaky rectified linear unit function (RRelu) is used instead of sigmoid activation function because it avoids vanishing gradient. Meanwhile, it is faster to achieve optimal weight. Other factors remain the same as the two-layer neuron network.

Then, this paper compares the classification accuracy of optimal two-layer network and optimal three-or-more layer network. The network with the best classification accuracy is chosen as the default model in the feature selection.

2.2 Feature Extraction by Using Autoencoder

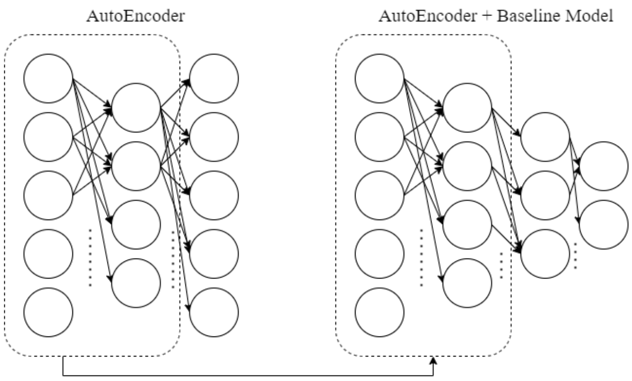
FOF's result shows that the baseline model performs inferior when classifies 'deceive or truth' response, but the model after feature selection results in a better model classification accuracy. It indicates poor feature extraction competence of the model. Autoencoder achieves excellent performance in extracting features [8, 9]. And it presents a better reconstruction of handwritten digit images than the principal component method [11]. The neuron network model may have better classification result when high-level features are pre-extracted. Thus, this paper employs autoencoder techniques to enhance the model classification result.

Firstly, this paper trains a single layer autoencoder. Then, this paper attaches the input layer and inner layer to the baseline model, as Figure 2 shows. The inner neuron numbers with the best classification result are selected as the optimal model. The number of training epochs and the learning rate are tuned in a way similar to the baseline model.

3.2 Functional Measure Feature Selection

There are multiple methods to evaluate the contribution of input to the neuron network model, including magnitude measures, functional measures and sensitive analysis. Functional measures perform best in determining the contribution of inputs to outputs and pruning [6]. The distinctiveness analysis is one of the functional measures which compares the similarity by calculating angles between multi-dimensional vectors [5]. The closer value of the angle is to 90, the higher distinctiveness between the two inputs, which leads to a more significant functionality contribution [7]. The distinctiveness of each pattern input is determined by the multiplication of input values and the aggregation weights of

Figure 2: Neuron Network Model with Autoencoder



input to hidden neurons. Overall distinctiveness is calculated by measures the angle between feature vectors. This paper uses the following formula to create the input contribution matrix.

$$\arccos(i,j) = \frac{sact(i,p)\cdot sact(p,j)}{\|sact(i,p)\|\|sact(p,j)\|} \tag{1}$$

$$\text{Where } sact(p,i) = value(p,i)\cdot \sum_{h}^{hidden\_neuron\_num} weight(i,h) \tag{2}$$

Function $\arccos(i,j)$ calculates the similarity between *i*th feature and *j*th feature across all patterns. Where $sact(p,i)$ counts the input contribution of each pattern's each feature. Variable p is the pattern index and variable i is the feature index. Variable $value(p,i)$ is the value at *p*th pattern's *i*th feature, variable. Variable $weight(i,h)$ presents the weight from *i*th feature to the *h*th hidden neuron of the neuron network model. So, $\sum_{h}^{hidden\_neuron\_num} weight(i,h)$ is the total weight of a feature to all hidden neurons. Then, $value(p,i)$ and $\sum_{h}^{hidden\_neuron\_num} weight(i,h)$ are multiplied to get the contribution of a particular feature of a pattern.

The distinctiveness of a feature is presented by the sum of angles between the feature and all other features. Features with the least distinctiveness are discarded in the neuron network. This paper repeatedly dumps the least distinctiveness feature in the remaining model input until the best classification result is achieved.

3.3 Leave One Out Cross Validation

This paper uses a small dataset. The model performance is greatly affected by the patterns that are select to build the model. The maximum number of patterns should be used to construct the model to improve model stability. So, this paper uses the leave one out cross-validation (LOOCV) to count model performance.

The dataset is separated into validation set and train set. The validation set randomly selects 3 deceive patterns and 3 truth patterns, which counts to about 20% of the whole dataset. Train set contains the rest of the data. For each run, the train set averages each cross's classification accuracy of LOOCV. The best model the chosen base on the LOOCV training result. Then, the model is tested on the validation set to achieve the final classification accuracy. For all training, the model is backpropagated for each epoch.

4 Result and Findings

4.1 Classification Accuracy Regarding One Direct Question

This paper tests the 'deceive or truth' classification capability by building a neuron network model. Table 2 demonstrates the resulting classification accuracy regarding different neuron network structures. The accuracy of a network model is the average classification accuracy of ten runs. The optimal model is the 2 hidden layer neuron network with 13 neurons in layer-1 and 6 neurons in layer-2 as the optimal network. The learning rate of the model is 0.001, and the number of optimal training epoch is 200. The optimal model achieves the average classification accuracy of 64%. It indicates that the 'deceive or truth' response of single questions can be modelled by using twenty ROI connection inputs. In addition, the 'deceive or truth' classification accuracy of this neuron network model is higher than FOF's models. The higher model identification capability may be a result of the neuron network model's better feature extraction power.

Table 2. Classification Accuracy of Different Neuron Networks Structure

No. Layers	No. 1st Layer	No. 2nd Layer	Accuracy (%)	No. Layers	No. 1st Layer	No. 2nd Layer	Accuracy (%)
1	3		34.80	2	10	6	61.60
1	6		48.40	2	13	6	64.00
1	9		46.40	2	16	6	56.80
1	12		50.00	2	19	6	60.00
1	15		56.40	2	10	9	60.80
1	18		50.40	2	13	9	61.60
2	4	3	57.60	2	16	9	57.60
2	7	3	59.20	2	19	9	59.20
2	10	3	63.20	2	13	12	60.80
2	13	3	60.00	2	16	12	69.00
2	16	3	60.80	2	19	12	60.00
2	19	3	55.20	2	16	15	53.60
2	7	6	59.20	2	19	15	60.00

4.2 Autoencoder Enhances 'Deceive or Truth' Identification Capability

Then, this report constructs a one-layer autoencoder to enhance the classification. This paper tests the two structures of autoencoders, with 10 hidden neurons and 15 hidden neurons. The loss of two autoencoders are listed in Table 3. 1500 training epochs is the optimal epoch of both autoencoders. The learning rate used to train the autoencoders is 0.001.

Table 3. Training Loss of Two Autoencoders

No. Hidden Neuron	No. Epochs 100	300	600	900	1200	1500	1800	2100	2400	2700
10	0.080	0.058	0.051	0.047	0.044	0.042	0.043	0.043	0.042	0.043
15	0.203	0.154	0.123	0.112	0.094	0.092	0.094	0.093	0.094	0.095

The 'deceive or truth' identification accuracy of two autoencoders is listed in Table 4. The 15 hidden neuron autoencoder shows a better result. In optimal epoch (i.e., 400) and learning rate (i.e., 0.001), classification accuracy reaches 76% before pruning. In FOF, the model accuracy improved after feature selection. This indicates the poor feature extraction capability of the baseline model. Autoencoder is one of the feature extraction method. The significant improvement of 'deceive' identification accuracy confirmed autoencoder's feature extraction capability.

Table 4. Classification Accuracy of Different Neuron Networks Structure

Autoencoder Neurons	No. 1st Layer	No. 2nd Layer	No. Epoch 100	400	600	800	1000	1200	1400	1600	1800	2000
10	13	6	52	60	64	64	48	68	52	68	60	70
15	13	6	64	76	64	60	56	60	72	52	68	68

4.3 Input Pruning Further Improves Classification Result

The angles between ROI connections in the optimal network are listed in Table 5. By repeatedly removing the least distinct input in the model, the model achieves the best results when the four most functional inputs (feature indices are 4, 12, 13, 17) are left in the network. The prediction capability of pruned model jumps to 82.67% on average. This validates the functional contribution analysis. In addition, the successful use of the technique shows that 4, 12,13 and 17 ROI connections are valid indicators of 'deceive or truth' response with single straightforward questions while others are noise or redundant information. The change of classification accuracy while pruning is shown in Figure 3.

Then, this paper tests the input pruning's affection on the autoencoder model. By randomly training the baseline model, baseline with input pruning model, feature extraction model and feature extraction with input pruning model five times, the classification accuracy on the validation set of each run are presented in Figure 4. The result shows that the input pruning can further improve the 'deceive' identification power after feature extraction as the average classification accuracy improved from 69% to 79%. But the overlay of two methods may not always performs better than the single input pruning method. This may because the input pruning already extracted the most distinct features, the extra feature extraction of autoencoder adds noise to the baseline model with input pruning.

5 Discussions

‘Deceive or truth’ response of a single question can be predicted by aggregation of twenty ROI connections as the classification accuracy is 65%. This paper also finds that autoencoder is a statistically effective method to extract features and enhance the model’s classification power. FOF finds that forehead-periorbital, chin-forehead, chin-perinasal and periorbital-forehead are features that are meaningful in machine learning models and statistics and sensible in physiology. This report finds four statistically significant features regarding the neuron network model as well. But it is impossible to compare the selected feature of this paper and FOF because the dataset features are not labelled. In addition, applying input pruning after feature extraction can further improve feature extraction’s classification accuracy.

This paper builds the model and analyzes model result regardless of the actual problem circumstance as there are no feature labels in the given dataset. This paper's neuron network creation and functional distinctiveness may not be valid in the actual problem circumstance. Therefore, functional distinctiveness should not be analyzed separately from the real problem circumstance. Further research should be conducted to rationalize the distinct features.

Theoretically, more information is provided when adding more features into a neuron network, and higher prediction capability is expected. However, the result of this paper contradicts the perception. Therefore, more studies may be taken to rationalize the inconsistency of low input information and high model performance.

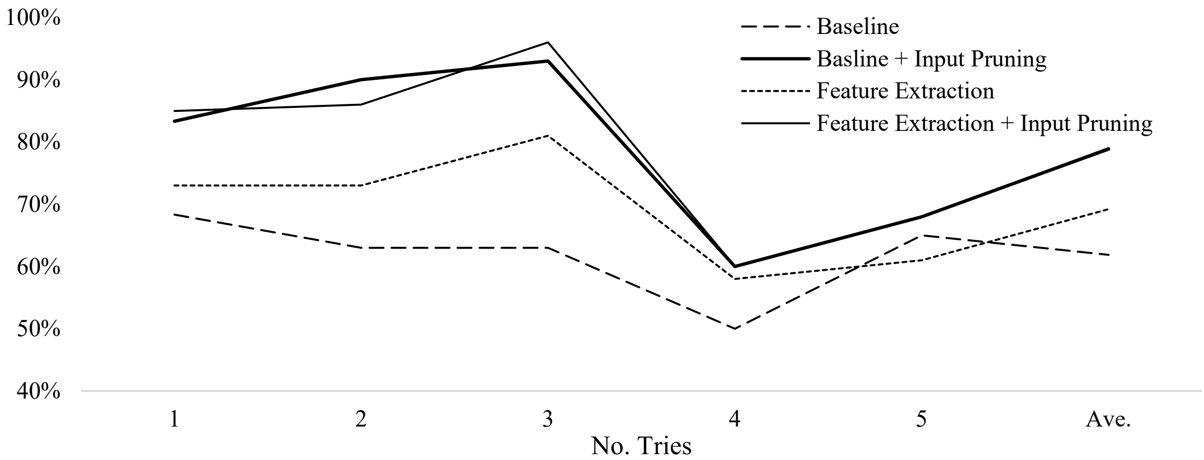
The model of this paper is not stable. The classification accuracy varies according to different initial weightings of the neuron network model and the patterns included in the validation set. Further research may focus on increasing classification stability.

Table 5. Angles between ROI Connections

Input Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0.0	0.0	0.0	0.0	23.2	25.0	0.0	0.0	0.0	0.0	15.2	0.0	34.9	22.7	0.0	19.0	0.0	30.8	0.0	18.8
1	0.0	0.0	0.0	17.1	25.0	24.6	17.5	19.2	17.3	0.0	0.0	0.0	36.4	18.9	0.0	22.7	17.0	30.4	0.0	22.4
2	0.0	0.0	nan	0.0	18.2	19.2	0.0	0.0	0.0	0.0	0.0	0.0	41.6	16.5	19.3	24.1	18.8	35.6	16.7	25.5
3	0.0	17.1	0.0	0.0	25.5	28.1	16.3	0.0	0.0	18.7	18.6	17.4	34.5	24.6	17.5	21.1	0.0	32.6	0.0	19.6
4	23.2	25.0	18.2	25.5	0.0	0.0	16.1	24.1	23.0	25.1	18.8	23.7	56.2	0.0	31.6	39.0	31.0	50.7	30.3	39.7
5	25.0	24.6	19.2	28.1	0.0	0.0	0.0	26.0	24.9	25.3	17.6	23.5	56.8	15.5	33.4	40.4	32.7	50.2	31.6	42.0
6	0.0	17.5	0.0	16.3	16.1	0.0	0.0	0.0	15.6	17.8	0.0	16.6	46.5	16.4	24.2	31.0	23.0	41.8	21.7	31.1
7	0.0	19.2	0.0	0.0	24.1	26.0	0.0	0.0	0.0	20.2	19.5	19.5	38.8	23.2	21.4	23.7	17.8	35.6	17.4	23.4
8	0.0	17.3	0.0	0.0	23.0	24.9	15.6	0.0	0.0	0.0	17.6	15.1	36.2	22.9	17.8	20.3	0.0	31.6	0.0	21.3
9	0.0	0.0	0.0	18.7	25.1	25.3	17.8	20.2	0.0	0.0	0.0	0.0	35.0	23.4	0.0	19.9	15.1	29.1	0.0	20.7
10	15.2	0.0	0.0	18.6	18.8	17.6	0.0	19.5	17.6	0.0	0.0	0.0	44.0	0.0	20.4	28.1	22.5	38.1	21.6	29.4
11	0.0	0.0	0.0	17.4	23.7	23.5	16.6	19.5	15.1	0.0	0.0	0.0	36.2	18.6	0.0	22.9	16.6	31.0	15.3	22.6
12	34.9	36.4	41.6	34.5	56.2	56.8	46.5	38.8	36.2	35.0	44.0	36.2	0.0	52.4	27.5	22.1	26.5	0.0	28.1	18.7
13	22.7	18.9	16.5	24.6	0.0	15.5	16.4	23.2	22.9	23.4	0.0	18.6	52.4	0.0	28.8	37.6	30.1	47.2	29.4	38.1
14	0.0	0.0	19.3	17.5	31.6	33.4	24.2	21.4	17.8	0.0	20.4	0.0	27.5	28.8	0.0	0.0	0.0	22.9	0.0	0.0
15	19.0	22.7	24.1	21.1	39.0	40.4	31.0	23.7	20.3	19.9	28.1	22.9	22.1	37.6	0.0	0.0	0.0	0.0	15.2	0.0
16	0.0	17.0	18.8	0.0	31.0	32.7	23.0	17.8	0.0	15.1	22.5	16.6	26.5	30.1	0.0	0.0	0.0	22.1	0.0	0.0
17	30.8	30.4	35.6	32.6	50.7	50.2	41.8	35.6	31.6	29.1	38.1	31.0	0.0	47.2	22.9	0.0	22.1	0.0	23.9	17.4
18	0.0	0.0	16.7	0.0	30.3	31.6	21.7	17.4	0.0	0.0	21.6	15.3	28.1	29.4	0.0	15.2	0.0	23.9	0.0	0.0
19	18.8	22.4	25.5	19.6	39.7	42.0	31.1	23.4	21.3	20.7	29.4	22.6	18.7	38.1	0.0	0.0	0.0	17.4	0.0	0.0
Sum of Dist.	189.6	268.4	235.6	291.6	501.1	516.8	335.6	330.0	263.6	250.4	311.4	278.8	672.5	466.2	264.8	387.0	273.2	570.9	251.2	390.7

Figure 3. Change of Classification Accuracy while Iteratively Removing the Least Distinctiveness Input

Figure 4. Change of Classification Accuracy while Iteratively Removing the Least Distinctiveness Input



References

1. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection, ACM, pp. 60 (2007)
2. Ben-Shakhar, G.: A Critical Review of the Control Questions Test (CQT). In: Kleiner, M. (ed.) Handbook of Polygraph Testing, pp. 103--126, Academic Press, London (2002)
3. Comert, Z., Kocamaz, A.F.: Comparison of Machine Learning Techniques for Fetal Heart Rate Classification. In: Acta Physica Polonica A, pp.132. 451--454. 10.12693/APhysPolA.132.451 (2017)
4. Derakhshan, A., Mikaeili, M., Nasrabadi, A. M., Gedeon, T.: Network Physiology of "Fight or Flight" Response in Facial Superficial Blood Vessels. In: Physiological Measurement. Physiological measurement, 40(1), p. 014002 (2018)
5. Garson, G. D.: Interpreting Neural Network Connection Weights. AI Expert, 6(4), pp. 47--51(1991)
6. Gedeon, T. D.: Data Mining of Inputs: Analysing Magnitude and Functional Measures. In: International Journal of Neural Systems, Vol 80, pp. 209--218 (1997)
7. Gedeon, T. D.: Lecture 4: Input Coding Techniques, Lecture Notes, Neural Networks, Deep Learning and Bio-inspired Computing COMP8420, Australian National University (2021)
8. Gedeon, T. D., Harris, D: Network Reduction Techniques. In: Proceedings International Conference on Neural Networks Methodologies and Applications, vol. 1, pp. 119--126, AMSE, San Diego (1991)
9. Meng, Q., Catchpoole, D., Skillicom, D., Kennedy, P.J.: Relational autoencoder for feature extraction, pp. 364, IEEE (2017)
10. Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., Khovanova, N: Machine Learning for Predictive Modelling based on Small Data in Biomedical Engineering, vol. 48, no. 20, pp. 469--474, IFAC PapersOnLine, (2015)
11. Tan, C. C., Eswaran, C.: Reconstruction of handwritten digit images using autoencoder neural networks, pp. 465, IEEE (2008)