Facing Truth: Sequential Feature Selection using Constructive Autoencoders for Deception Classification

Ed Muthiah

Research School of Computer Science
Australian National University, ACT 2601
u5589751@anu.edu.au

Abstract. Feature selection is paramount in in machine learning. This paper introduces a novel 'wrapper-embeded' feature selection technique using cascade constructed autoencoder for sequential feature selection (AutoConSFS). The technique is applied to thermal facial analysis data to detect deception. The introduced technique is also compared with sequential feature selection using fully connected neural networks (FCNSFS). While both networks perform better using all available features, AutoConSFS (72.7%) performs 9.1% better than FCNSFS. However, the AutoConSFS feature selection was 10.1% below par compared to the original deception detection features selected manually using t-test, relative entropy, and Wilcoxon tests.

Keywords: Constructive Cascade, Autoencoders, Feature Selection, Bioinformatics, Deception, Facial Thermal Imaging, Neural Network

1 Introduction

1.1 Background

Feature Selection (FS) has been a long-term focus of the machine learning community. In the field of bioinformatics, well-implementation feature selection has shown to reduce dataset redundancy and model overfitting, whilst increasing interpretability and model performance by time and accuracy on regression and classification tasks [1]. Common FS techniques can be split into three categories - filter, wrapper and embedded methods. A comparison between the three is presented in both Figure 1 and Table 1 below.

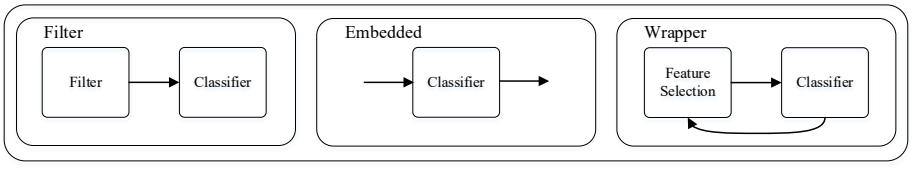

Fig. 1. Filter (left), Embedded (middle) and Wrapper (right)

Table 1. Comparison of Filter, Wrapper and Embedded feature selection.

	Filter Method	*Wrapper Method*	*Embedded Method*
Description	Classifier Independent Feature Statistics	Predictive Models	Incorporated in Classifier Design
Speed	Computationally Faster	Slower	Medium
Overfitting	Avoids Overfitting	Prone to Overfitting	Less Prone to Overfitting
Performance	May fail to select best features	Better Performance, Captures feature dependences	Good Performance, Classifier Dependent Selection
Examples	Correlation, Chi-Square, Info Gain	Forward Selection, Sequential Selection,	Autoencoders, NNs, Ridge Reg

1.2 Motivation

The motivation of this paper stems from combining the benefits of both wrapper methods and embedded methods in feature selection. Wrapper methods have good accuracy and explainability but limited speed and generalization. In contrast, embedded methods are high accuracy and generalizable but are black box.

Three past efforts form the reasoning behind the algorithm contributed by this paper:

(1) The application of neural networks (NNs) to wrapper-based methods for FS was successfully implemented by Onnia et al. [2], who implement a variation of the sequential feature selection (SFS) by adding one neuron (feature) to a multilayer perceptron (MLP) input layer at a time with one fixed size hidden layer. Upon testing on various simulated and real world problems, Onnia et al. showed that the neural network based wrapper methods reduced the size of the feature space significantly and improved classification accuracy compared to using all features.

(2) The concept of NNs for wrapper-based FS in (1) was extended in Ramjee et Al's efficient feature selection algorithm using autoencoders (AE). AEs are a common neural network architecture found in many deep learning techniques. Their structure as show in Figure 2 below aim to achieve dimensionality reduction of the input to a lower dimensional space before, targeting to reconstruct the input from this reduce feature space and can be considered an embedded filtering technique.

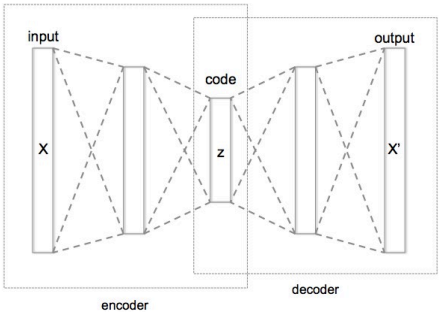

Fig. 2. Typical Autoencoder Structure. Source: Wikimedia Commons

Given the high compute of wrapper based techniques Ramjee et. Al implement an autoencoders to perform greedy backward elimination of features, removing features that are not critical to a classification task. In their AMBER strategy they train a *single* hidden layer autoencoder using sequential feature selection where they set the d^{th} feature of the training set to 0 in a round-robin fashion and retrieve both the MSE and classification accuracy to rank and remove k features.

Upon comparison to other feature selectors such Fisher, Conditional Mutual Information Maximisation (CMIM), R-value, Feature Quality Index (FQI) the AMBER strategy is shown to return higher accuracies at lower computational cost on popular datasets such as MNIST and RadioML. Regardless, the autoencoder parameters such as hidden layer depth and width are chosen empirically rather than through self-construction.

(3) The Cascade Correlation (CasCor) algorithm developed by Fahlman et al. is considered to be a pre-cursor to deep neural networks (DNNS) and boosting [4]. The architecture self-constructs itself by beginning with a minimal NN with a single input layer is connected directly to the output layer and cascade one hidden neuron at a time. Alongside being connected to the outputs of previous hidden neurons, each hidden neuron also maintains its connection with the input layer. The selection hidden neurons achieved using a candidate pool which aims to maximize the correlation between the neuron's output and the remaining network error. These layers are then frozen resulting in a network with permanent feature sets that are computation more efficient than backpropagation and inherently lean. A visualization of the cascade correlation architecture is presented in Figure 2.

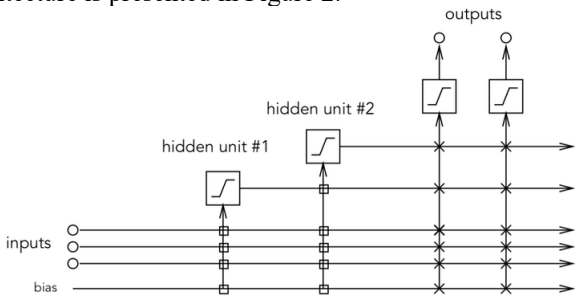

Fig. 3. Fahlman et al.'s Cascade Correlation learning architecture [4]

1.3 Contribution

This paper extends the efforts of neural-network based SFS methods in (1) and (2) and combine it with the principles of (3) to contribute a:

- A constructive cascade autoencoder network for sequential feature selection (AutoConSFS) algorithm (implemented 2 hidden layers).
- SFS using a fully connected neural network with 1 fixed hidden layers (FCNSFS).
- A variety of comparisons to using full feature sets, selected feature sets and compared to statistical methods such as t-test, relative entropy, receiver operating characteristic, and Wilcoxon test.

1.3 Dataset

To validate the AutoConSFS and FCNSFS algorithms, data from the paper *"Network physiology of 'fight or flight' response in facial superficial blood vessels"* by Derakshan et al. is used [6]. Derakshan et al.'s paper presents a novel framework to identify the dynamic pattern of blood flow changes in the cutaneous superficial blood vessels of the face for 'fight or flight' responses through facial thermal imaging.

The data is from a mock crime scenario, in which 31 participants were divided into two groups: deceptive and truthful. All participants were brought to an interrogation room and asked the question: "Did you steal the gold necklace?" During which, psycho-physiological effects on their faces were captured using a thermal camera. Within each thermal image, five regions of interest of the face were defined as the periorbital, forehead, cheek, perinasal and chin as shown in Figure 2.

Fig. 4. Five facial ROIs defined as the periorbital PO (1), forehead FH (2), Cheek CK (3), perinasal PN (4) and chin CN (5) [6]

The blood flow between facial regions was assumed to be a time series. Thus, Derakshan et al. used an extension of the standard Granger Causality (eGC), which shows the causal relationship between two time series, to determine the relationship between paired facial regions.

Table 2. Paired eGC facial region index. Manually selected features in bold [5]

	periorbital	***forehead***	***cheek***	***perinasal***	***chin***
periorbital		**1**	2	3	4
forehead	**5**		6	7	**8**
cheek	9	10		11	12
perinasal	13	14	15		**16**
chin	17	19	19	20	

Upon feature selection using t-test, relative entropy, receiver operating characteristic, and Wilcoxon test. The bolded ROI pairs (shown in Table 2) were manually selected by Derakshan et al. for returning the best classifier performance. Specifically, the values correlate to PO-FH, CN-PN, FH-PO and CN-FH. These selected features provide a benchmark for CasCorSFS and FCNSFS implemented in this paper. The eGC value dataset contained deception results from 31 suspect each with 20 facial pair features (as shown in Table 2). The dataset contain 15 liars (encoded as True) and 16 non-liars (encoded as False).

2 Method

2.1 Sequential Feature Selection Algorithm

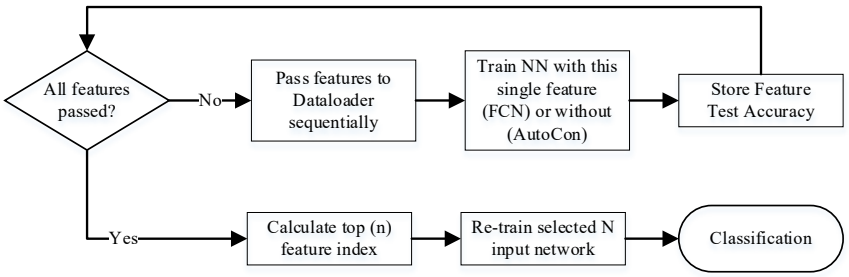

Fig. 5. Sequential Feature Selection (SFS) Algorithm Implementation

Figure 3 above captures the implementation of the SFS algorithm. This technique is applicable to both the AutoConSFS and FCNSFS architecture outline in section 2.3. Note that for FCN only one feature was used at one time for training the network, while for AutoCon one feature was removed, leaving only 19 features.

2.2 Dataset Preparation

- The GC indexes were normalized to [0 1] for each person using the Min-Max normalization technique.
- From the 31 total examples, the dataset was split to provide 20 or training 10 examples of liar/non-liar and 11 images for testing with 5 liars and 6 non-lars
- A consistent batch size of 5 examples sampled from a shuffled dataset was used throughout AutoConSFS and FCNSFS training.

2.3 Network Architecture and Training

2.3.1 Fully Connected (FCN) SFS and Network

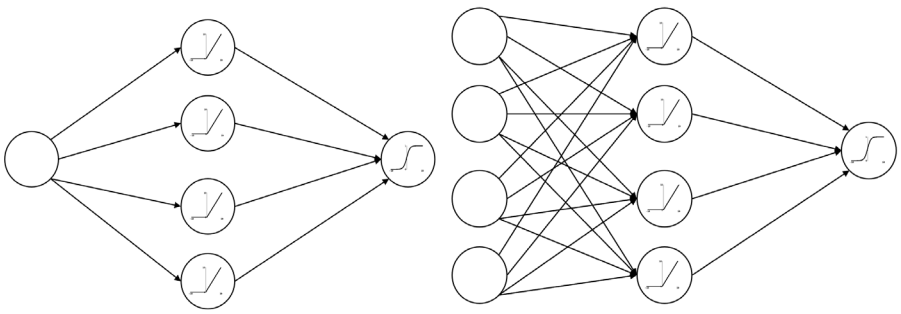

Fig. 6. Fully Connected Neural Network (FCN) with 4-unit hidden layer.

Left image shows FCN during single FS and right image shows after top 4 are selected. The FCNSFS was trained using FCN with a 1-4-1 architecture. The hidden layer used a ReLU activation with a dropout of 0.2. The output layer used a Sigmoid activation. The optimizer was Adam, with a 1e-3 learning rate and L2 regularisation of 1e-5 for generalization.. Training was run for 500 epochs. This network architecture was chosen because of the low number of training and testing samples. Given the sequential selection process of testing one feature at a time only one input node is required, it was found that there was little gain using more than one hidden layer with a single input and was likely to result in local minima.

2.3.2 Constructive Autoencoder (AutoCon) SFS and Network

The AutoCon SFS aims to combine the benefits of both wrapping and embedded filtering techniques whilst also self-construcing a multi-layer AE. To achieve this AutoCon methodology has four distinct steps:

(1) Base AE definition (20-k-20) - Use all features (20) through a base single hidden layer autoencoder with (k) hidden units and use the MSE to determine the optimal number ($k_{optimal}$) of hidden neurons in the base AE.

The AE is trained for 10 epochs, with an input size of 20 and k hidden size that iterates from 1 to 19. We cap at 19 to ensure that hidden layer size is smaller than the input layer to achieve information loss. Given, that we only have 20 training samples, a relatively low number of training epoch was used for early stopping. The encoder and decoder both use ReLu as the activation function. SGD optimizer was use with learning rate 0.3 and momentum 0.9 alongside with a MSE loss.

(2) Base AE Sequential Features Selection (19 - $k_{optimal}$ - 19) – In a similar approach to Ramjee et. Al in [5] we train the AE with one removed feature column (leaving 19 features) in a round robin fashion. The MSE is then used to rank the top 10 input features. The value of 10 here was chosen empirically as 50% of the input features. This is because too many features would not allow for dimensionality reduction while too few features would leave only a few (if not single) neuron widths in further hidden layers. The Base AE SFS is trained for 10 epochs, with an input size of 19 and $k_{optimal}$ hidden size. The encoder and decoder both use ReLu as the activation function. SGD optimizer was use with learning rate 0.3 and momentum 0.9 alongside with a MSE loss.

(3) SFS AE Classification (10 - $k_{optimal}$ – 10) – Upon establishing this initial SFS AE with a single hidden layer from Step 1and Step 2, we train the AE with 10 epochs, input size 10, hidden size $k_{optimal}$. The encoder and decoder both use ReLu as the activation function. SGD optimizer was use with learning rate 0.3 and momentum 0.9 alongside with a MSE loss. Once training of the AE feature selector is complete we freeze the AE weights up to the bottleneck layer, remove the decoder and append a classifier layer with input size $k_{optimal}$ and output size 2. The output size correspond to the categories 0

and 1 for not-deception and deceptive. The classifier layer uses a LogSoftmax loss. The classifier layer is then trained with 40 epochs, negative log likelihood loss, and adam optimizer with learning rate 0.3.

.

(4) AutoConSFS Classification (10 $k_{optimal}$ – $k_{bottleneck}$ - $k_{optimal}$ – 10) – To understand the the 4th Step of AutoCon we consider the similarities between Cascade Correlation and AutoCon as shown in Table 3.

Table 3. Similarities between Cascasde Correlation [6] and AutoCon

Cascade Correlation	***AutoCon***
Begin with a minimal NN	Begin with 1 hidden layer AE from Step 3.
Cascade train one hidden neuron at a time.	Cascade train one encode-decode at a time.
Freeze previously trained neurons.	Freeze previously train encode-decode layers.
Candidate pool for hidden neuron width based on correlation.	Candidate pool for bottleneck width based on classification accuracy.

Thus the to build AutoCon we first repeat the first half Step 3, however once the training is complete of the base SFS AE we freeze the both the encoder and decoder layer weights. We then add a stack an encode-decode layer in-between these for training. The number of units ($k_{bottleneck}$) in the new bottleneck second layer is self-determined using a candidate pool comparison with classification accuracy. The values of $k_{bottleneck}$ range from [1, $k_{optimal}$] to ensure dimensionality reduction. We train the second encode-decode pair 10 epochs, ReLu activation, SGD with lr 0.3 and momentum 0.9 and MSE loss.

Finally to perform classification in a similar manner to the previous step we freeze all weights up to the bottle neck and train the classifier with the same parameters as in Step 3. It is hypothesized that the layer-wise pre-training will allow for more permanent feature sets that are computationally more efficient. A visualization of the AutoCon architecture Step 4 is shown in Figure 7.

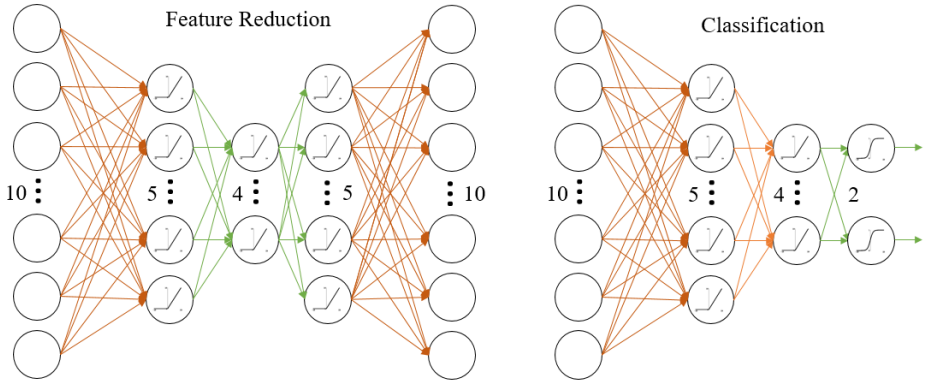

Fig. 7. AutoCon Step 4 reduction and classification phases with frozen weights in orange and learnable weights in green.

2.4 Hyper Parameters Optimisation

Numerous hyperparameter experiments were conducted for both the FCN and AutoCon networks including:

- Different activations: tanh, sigmoid, relu
- Learning rates: 3e-1, 3e-2, 3e-3, 3e-4, 3e-5
- Regularisation strengths: 1e-3, 1e-4, 1e-5
- Number of epocs:1,10,40, 50, 100, 250, 500, 1000
- Dropout: 0.1, 0.2, 0.25, 0.5
- Number of nodes in FCN hidden layer: 1,2,4,8,16
- Number of input features to AutoCon: 20,15,10,5,3

3 Results and Discussion

The results obtained from implementation can be summarized into 2 categories, FCNSFS results and AutoConSFS results. Relevant comparisons to the original paper are made in line in each section.

3.1 Fully Connected Network Sequential Feature Selection Results

Figure 6 below shows the results of training the FCN with all 20 feature inputs. Compared to the training results of the top for 4 selected values.

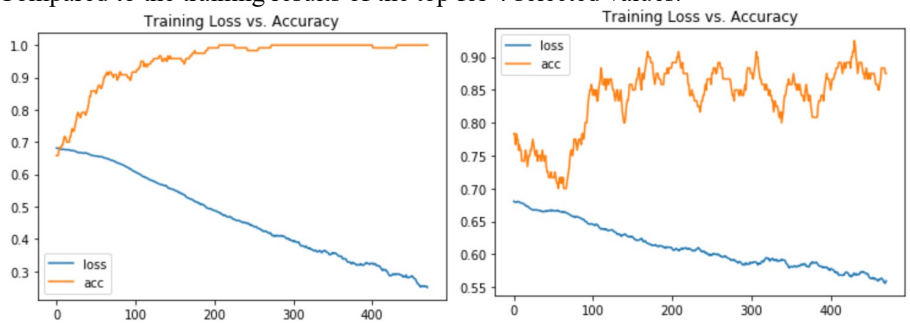

Fig. 8. Fully Connected Neural Network training results using all 20 features (left) and top 4 selected features (right)

The test accuracy for pre and post feature selection was 54.5% and 63.6% respectively. Subsequently the FCNSFS network performs nearly 10% better using the NN based feature selection. The pre feature selection accuracy is almost identical to Derekshan et al who posted ~54%, however the Derekshan FS techniques return a significantly better accuracy at 83.3%. The top four facial ROI pair returned by the FCNSFS network were 1. Cheek-Periorbital, 2. Perinasal-Chin, 3. Periorbital-Forehead, 4. Cheek-Perinasal. Of these, only Periorbital-Forehead Matched with the statistical feature selection conducted by Derakshan et al.

From the training graphs in Fig 6 we see that although the feature selected network (right) has a more fluctuating loss curve, it does not overfit the dataset like the full feature network, which quickly saturates to 100% accuracy.

Constructive Autoencoder Sequential Feature Selection Results

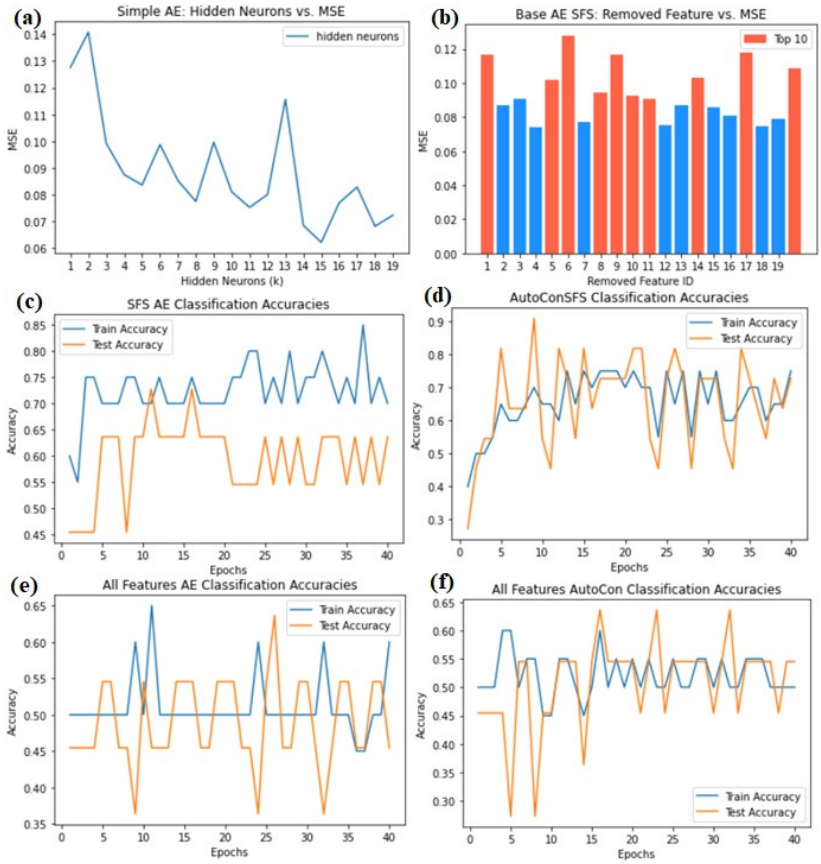

Fig. 9. AutoCon Results: (a) Simple AE hidden neuron selection, (b) Input SFS, (c) SFS AE classification, (d) AutoConSFS classification, (e) All Features AE classification, (f) All Features AutoCon classification

Figure 9 above shows extensive results from the implementation of the methodology:

(a) Initially, when determining the optimal base AE hidden neurons based on a MSE, (a) shows that the optimal is around 5 neurons. Althought the MSE continues to decrease with more neurons as expected, the information loss is lower and lower as the bottleneck layer increases until the mapping is essentially an identity. Thus as a trade between accuracy and dimentionality reduction 5 neurons is selected as $k_{optimal}$ leading to a 20-5-20 AE.

(b) In sequential feature selection phase one feature was removed in a round robin fashion and (b) shows the impact on the MSE. It is assumed here that features with lower MSE when remove have redundancies as thus the top 10 features are selected. During the SFS the results selected features 1 (PO-FH), 5 (FH-PO), and 8 (CN-FH) which correspond directly to also ¾ features selected by Derakshan et al.

(c, d) From (c) and (d) we see that with a single hidden layer AE, using SFS we achieve a train/test accuracies 0.70/0.63, while using an additional hidden layer and the AutoCon technique we achieve improved train/test accuracies of 0.75/0.72. This improved accuracy is likely a result of the deep autocon structure which allows for more non-linear representations in the data to be captured that a single hidden layer. Early stopping at 40 epoch was a key decision here as for greater values the small dataset began to overfit to the training data. When comparing with the original Derekshan et al paper we see that commendable AutoConSFS 72% test accuracy falls short of the 83.3% reported by Derekshan et al.

(e, f) In (e) and (f) we see a significant different that the SFS technique has when compared to simply using all available features. With a single hidden layer AE, using SFS we achieve a train/test accuracies 0.60/0.45, while using an additional hidden layer and the AutoCon technique we achieve improved train/test accuracies of 0.5/0.55. Thus, it is evident that autocon straight fails to learn when all features are used. Although initially, this may look to be a flaw of the model when compared to the Derekshan paper, the reported full feature accuracy was also only ~54%.

As a final comparision, we find that AutoConSFS significantly improves on the FCNSFS train and test accuracies by 9.1%. Of the selected features, only Periorbital-Forehead was common across FCNSFS, AutoCONSFS and Derekshan. However, the test accuracy of AutoConSFS suggests other descriptive features were also identified.

4 Conclusion and Future Work

The main goal of this study was to implement a novel wrapper-based feature selection algorithm using fully connected and constructive autoencoder neural networks. In the application of these techniques to deception detection, the use of top n feature selection significant and consistently improved the performance when compared to using all available features. Although this result is promising, the choice of a two stack autoencoder was only a proof of concept and thus future implementations should experiment with greater network depths and widths. An immediate future task could be to assess feature selection with a joint metric that combines that optimizes dimensionality reduction and classification error.

References

1. Saeys, Yvan, Inaki Inza, and Pedro Larranaga. "A review of feature selection techniques in bioinformatics." bioinformatics 23.19 (2007): 2507-2517.
2. Onnia, Vesa, Marius Tico, and Jukka Saarinen. "Feature selection method using neural network." 2001 ICIP (Cat. No. 01CH37205). Vol. 1. IEEE, 2001.
3. Fahlman, and Scott E. The cascade-correlation learning architecture. CMU E, 1990.
4. Kabir, Md Monirul, Md Monirul Islam, and Kazuyuki Murase. "A new wrapper feature selection approach using neural network." Neurocomputing 73.16-18 (2010): 3273-3283.
5. Ramjee, Sharan, and Aly El Gamal. "Efficient wrapper feature selection using autoencoder and model based elimination." arXiv preprint arXiv:1905.11592 (2019).
6. Derakhshan, Amin, et al. "Network physiology of 'fight or flight' response in facial superficial blood vessels." Physiological measurement 40.1 (2019): 014002.