# Using Weight Matrix Analysis for Data Mining of Inputs and Applying 1D-CNN for Optimization

Tian Wu1

Research School of Computer Science Australian National University, Canberra Australia <u>tian.wu@anu.edu.au</u>

**Abstract.** Real-world data sets often contain much redundant and irrelevant information which makes the information quality of the inputs compromised to some extent. An effective way to select valuable features over irrelevant information is to make use of the weight matrix analysis. This paper utilizes a dataset composed of human physiological signals from observers which are used to detect presenters' subjective belief in the information they present. After conducting the weight matrix analysis, we are able to find out inputs of large magnitude but redundant to the outputs in the data mining process. After removing these redundant inputs, we use both 2D and 1D-CNNs to train the model and improve the accuracy. This paper presents a new application for first adopting the magnitude analysis to rule out inputs of least significance, and then applying 2D and 1D-CNN to train the models to improve performance. In this way, we are able to keep the valuable and unique inputs and train our model with higher efficiency and accuracy.

Keywords: Weight Matrix, Data Mining, 1D Convolutional Neural Network (1D-CNN), CNN

# **1** Introduction

The pupil dilation information is used as effective indicators to measure people's mental state and activities. According to researchers [1], pupils will become significantly larger after positive or negative stimuli than after a normal stimulus. Thus the pupillary dilation data along with other physiological signals can be used in an emotionally engaging event in order to detect people's true feelings even if they want to verbally or physically hide them. The first step we take is to devise an effective neural network to investigate the problem thoroughly by reproducing the similar results described in the original paper related to this physiological signals dataset [2]. The network devised is a fully connected neural network of one hidden layer trained with a loss function for backpropagation. A method is also developed to evaluate the performance of the neural network we just created. After the implementations of the neural network, we conduct a weight matrix analysis to better understand the importance of each of the input features. According to researchers, by combining the weight matrix obtained from both the inputs to hidden neurons and hidden neurons to the outputs, we can calculate a final Q matrix to measure the magnitude of the contributions from the inputs to the outputs directly [3]. By removing the least significant input features, we can train our neural network while saving computational resources and still maintaining the same level of accuracy. Convolutional neural networks (CNNs) are used by researchers to construct feed-forward Artificial Networks (ANNs) by adding convolutional and subsampling layers to learn hidden and complicated patterns efficiently [4]. Therefore, we can restructure the dataset to a two-dimensional representation and utilize a 2D-CNN to optimize our model and improve the accuracy. Recent studies have also shown that 1D-CNN can be used to classify time series signals effectively [5]. The dataset we use in this paper is composed of physiological signals captured by sensors while participants were watching the videos during the experiments, so we can further take full advantage of the convolutional neural network by re-training our dataset with 1D-CNN model with less computational resources and higher efficiency.

# 2 Method

#### 2.1 Model Description and Data Preprocessing

The original dataset includes multiple physiological signals but we mainly focus on the pupil dilation data as is used by the researchers [2]. First, we build a neural network which contains a hidden layer of 100 using sigmoid as the activation function. The output layer has two neurons while the inputs are extracted from the physiological signals of pupil dilations which contain 39 features. In this way, our main data is aligned with the original dataset, and we can check our model against their original benchmark performance. Before using the raw data, we also need to normalize the extracted pupil dilation data to reduce the side effect of individual bias because the variations of the pupil dilation data are more important than the absolute values of individual pupil sizes.



Fig. 1. General structure of our neural network with one hidden layer and two outputs

The neuron network is trained with the Adam optimizer without further adaptations. We use the CrossEntropy loss as our loss function for backpropagation. For effective cross validation, k-fold is often preferred when the dataset is relatively small. However, for our particular dataset, it is worth pointing out that continuous physiological data with multiple data points from the same person can reflect his/her responses to a stimulus. Therefore, random splitting the dataset is replaced with a more carefully designed selection method to ensure that a participant's data is not mixed in both the test and the training set. The method treats one participant's data as a whole and is used as the test set while the rest form the training set and are used for training. We repeat this method for all participants to make full use of the dataset. Finally, to measure the performance of our network, we break it down to two areas of percentages consisting of accuracies for the doubt conditions and trust conditions separately.

#### 2.2 Weight Matrix Analysis

Gedeon [3] puts forward an robust technique to measure the contributions from each input value to the output using the following formula:

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

where the  $P_{ir}$  and  $P_{rk}$  are matrice to measure the contribution of an input to a hidden neuron [6] and a hidden neuron to an output neuron [3], respectively:

$$P_{ij} = \frac{|w_{ij}|}{\sum\limits_{p=1}^{ni} |w_{pj}|},$$
(2)

$$P_{jk} = \frac{|w_{jk}|}{\sum\limits_{r=1}^{nh} |w_{rk}|}.$$
(3)

Compared with the measure proposed by Garson [7] and Milne [8], the Q matrix calculated is more robust since the signs of the contribution can be recovered using the raw data by simple statistical formulas. Essentially, we can calculate the top most significant inputs with relatively high stability. After training the network, we use two methods to test the stability and the effect of the top 3 most significant inputs. The first is to overtrain our network to gain the largest 3 values of our Q matrix. The second is to train our network without the 3 least significant inputs and compare the results of accuracies.

## 2.3 2D Convolutional Neural Networks

Researchers have commonly used the conversion techniques to reshape the 1D temporal signals into an  $n \times m2D$  matrix as an image [9]. This technique has the advantages of direct and convenient conversion with robust performance. The processed 2D images are then used for deep CNN models for training. 2D-CNNs usually require much hardware training resources in order to achieve reasonable generalization abilities, therefore low-memory devices such as mobile platforms may not be suitable for such applications [10]. After finding out the least significant input features, we can then trim these redundant inputs out, therefore saving computational resources for future modeling. After obtaining the relatively small size of our one dimensional time series of pupil dilation dataset, we can easily transition our one dimensional features into a small two dimensional image. Specifically, after leaving out the least three significant inputs, we convert all one dimensional inputs to  $6 \times 6$  images. At a high level, we construct our 2D-CNN consisting of two parts: first, a convolutional encoder including two convolutional layers; second, a dense multilayer perceptron block of three fully-connected layers.

# 2.4 1D Convolutional Neural Networks

Traditional CNNs are designed based on the 2D data from images and videos to discover and learn the hidden patterns and representations behind the scene. However, for time series classification, the dataset is focused on 1D signals such as various kinds of sound representations [11], raw audio waveforms [12], and ECG signals [13]. One of the most advantageous aspects of using a 1D-CNN model is that 1D convolutions require much less computational complexities compared with 2D convolutions. Meanwhile, 1D-CNN models normally have much fewer parameters under the size of ten thousand, while in contrast, a deep 2D CNN model can engage more than one million parameters in the architecture. To ensure that our 1D-CNN model can capture and learn the right patterns in the time-series data, we need to try out different kernel sizes and select the best fitted hyper-parameter. We adopt a grid search method in finding the proper sizes for kernels during a series of experiments. After selecting the suitable kernel size for the convolution, we apply two blocks of the convolution and pooling operations to extract and learn the potential patterns in the one dimensional date. After flattening the feature maps, we then utilize three fully-connected layers to capture the signals for more accurate classifications.

# **3** Results and Discussion

# 3.1 Reproducing the neural network

If an observer feels doubt in the process of a presentation, his/her pupil will respond with a higher level of arousal [14]. We train our neural network by using the pupillary responses to predict observers' doubting and trusting reactions. Our results show that the overall accuracy is a little above 50%. In more detail, in the doubt condition, the final accuracy is around 50% which is roughly at a chance level. While in the trust condition, on the contrary, the final accuracy can be as high as 66%, which is significantly higher than the chance level of 50%. Both accuracies, however, are higher than the verbal responses from the observers [2]. The results indicate that the trained neural network can classify the doubt and trust conditions better than conscious veracity judgement. This trained network serves as a baseline for our future improvements to check against.

# 3.2 Weight Matrix Analysis Results

# 3.2.1 Q Matrix Model

After fully training our neural network, the two weight matrices are utilized to calculate the final Q matrix to determine each input's significance on the output. By applying formula (1) mentioned earlier, we thus overtrain our neural network to get 3 most significant and 3 least significant input orders.

Q Model	Most significant	2nd most	3rd most	Least significant	2nd least	3rd least
Input Order	31	28	36	11	24	35

Table 1.	Significance	Levels	of inputs
Table T	Significance	101010	or inputs

#### 3.2.2 Stability of the contributions

By repeating the overtraining processes, we can check the robustness of the Q matrix model. The following figure shows that during the overtraining processes, the top 3 most of the significant and the insignificant inputs relatively keep their significance levels.



Fig. 2. Changes in the 8 most significant inputs during overtraining

#### 3.2.3 Training without the least significant inputs

Since the least significant inputs contribute little to the outputs while consuming the time and space in our training of neural networks, it is a good idea to leave them out altogether to train the network more efficiently, especially if we want to convert the inputs to higher dimensional data for convolutional networks. So we experiment without the three least significant inputs. In our case, we drop the 11th, 24th and the 35th inputs to re-train our neural network. The results show a slightly improved accuracy of 52% accuracy in total, with 47% accuracy for the doubt situation and 62% for the trust situation. Therefore, the neural network is still highly valid and the quality of the training remains largely unchanged. The greatest benefit is that we save time and computation space by removing insignificant inputs.

# 3.3 Using 2D-CNN in training

As is shown in Fig. 3, we construct a 2D-CNN model that consists of two parts: first, a convolutional encoder including two convolutional layers; second, a dense multilayer perceptron block of three fully-connected layers. The basic units in each convolutional block comprise a convolutional layer, a ReLU activation function, and a subsequent max-pooling operation. For each convolutional layer, we use a  $3 \times 3$  kernel and a ReLU activation function. In this way, we are able to map spatially arranged inputs to a number of two dimensional maps and increase the number of channels. We expand to 6 output channels first, then increase to 16 afterwards. We utilize the  $2 \times 2$  pooling with stride of 2 to reduce the dimensionality for downsampling. Usually, to smoothly transition the outputs from the convolutional encoder to the dense blocks, we need to flatten our data back to one dimension for dense layers. Finally, we can perform the MLP classification with 16, 50 and 2 outputs, with the last 2 outputs being the final result of classification. After training, surprisingly, the accuracy of doubt situation judgement improves to 61% at the highest with a mean value of around 55%, which is greater than the chance level, and it is much better than the 47% from the normal MLP model. However, the accuracy for the trust situation drops slightly to an accuracy level of around 58% on average. The total accuracy for identifying both doubt and trust scenarios improves slightly to around 56%.



Fig. 3. Configuration of the 2D-CNN with convolution and fully-connected layers

#### 3.4 Using 1D-CNN in training

Generally speaking, for 1D-CNN models, hyper-parameters mainly include four different aspects: first, the number of hidden layers and neurons which consist of hidden CNN layers and MLP layers; second, the kernel sizes for every CNN layer; third, subsampling factor for each CNN layer; finally, the functions we choose for pooling and activation operations. As illustrated in Fig. 4, our 1D-CNN consists of two distinct layer types: first, the CNN layers where we perform the 1D convolution, adopt a ReLU activation function, and conduct the max-pooling operations; second, the typical MLP of three fully connected layers. Intuitively, by expanding the channel sizes, the model can respond to different sets of features more effectively. Therefore, we increase the channel sizes to 6 and 16 respectively to try to learn the patterns as much as possible. However, the underlying mechanism is more complicated as some representations are not learnt independently, but rather optimized jointly to contribute to the overall performance. Therefore, by adopting three fully connected layers, we can finally translate what the model has learnt in the CNN architecture to a definite classification result. For convolutional layers, we select a  $3 \times 1$  and a  $5 \times 1$  kernel after a series of trials. On the other hand, the max pooling operations are conducted with  $2 \times 1$  and  $3 \times 1$  kernels, each with a stride of 2 to downsample the one dimensional data. Before transporting to fully connected layers, we perform the flattening operation as usual. We adopt a three-layer MLP model for effective classification. After training, we achieve a total accuracy of 57.8%, with standard deviation of 3.3% which is even better than the computationally intensive 2D-CNN model. For accuracy in doubt situations, our model can improve the result from worse than chance level (47%) to nearly 60% at best, while for trust situations, the accuracy drops slightly to around 58%. It is worth noting that the 1D-CNN drastically reduces the accuracy differences between the trust and doubt situations, which is not possible to achieve with simple MLP models.



Fig. 4. Configuration of the 1D-CNN with convolution and fully-connected layers

# 4 Conclusion and Future Work

In this paper, we build a fully connected neural network to detect the difference between doubt and trust conditions using the pupillary dataset. After training the neural network, we use the technique of weight matrix to determine the importance of each input to the output. By overtraining our network, we make sure that our model is robust and the significance level is stable. To further take advantage of the contribution matrix, we can save computational resources by leaving out the least three significant inputs while still keeping the accuracy. Moreover, we utilize the convolutional neural networks to further train and improve our models. By converting the one dimensional data to two dimensional image representations, we take advantage of the 2D-CNN to construct a deep neural network to learn the complex patterns of the data. Furthermore, we also apply the one dimension data directly to the 1D-CNN model which requires much less computational resources while maintaining the same or slightly higher level of accuracy. One direction worth more study in the future is to figure out the underlying optimization mechanism in applying the 1D-CNN model to one dimensional data, especially for time series datasets collected by different sensors. On the other hand, to find out the most fitted kernels for the convolutional networks, we mainly rely on a hand-crafted process of trying out different combinations of size matrices. This grid search approach in the procedure is time-consuming and prone to errors. More work needs to be done both in automatically selecting the hyper-parameters and evaluating the impacts at the same time. Researchers have recently proposed a novel architecture of Omni-Scale 1D-CNN (OS-CNN) for capturing appropriate kernel sizes [15]. This method can potentially improve the efficiency of our model during the learning period and offer baseline performance indicators for multiple applications.

# References

1. Caldwell, Sabrina, Tamás Gedeon, Richard Jones, and Leana Copeland. "Imperfect understandings: a grounded theory and eye gaze investigation of human perceptions of manipulated and unmanipulated digital images." In Proceedings of the World Congress on Electrical Engineering and Computer Systems and Science, vol. 308. 2015.

2. Zhu, Xuanying, Zhenyue Qin, Tom Gedeon, Richard Jones, Md Zakir Hossain, and Sabrina Caldwell. "Detecting the Doubt Effect and Subjective Beliefs Using Neural Networks and Observers' Pupillary Responses." In International Conference on Neural Information Processing, pp. 610-621. Springer, Cham, 2018.

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. Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." In 2017 International Conference on Engineering and Technology (ICET), pp. 1-6. Ieee, 2017.

5. Kiranyaz, Serkan, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. "1D convolutional neural networks and applications: A survey." Mechanical systems and signal processing 151 (2021): 107398.

6. Wong, Patrick M., Tamás D. Gedeon, and Ian J. Taggart. "An improved technique in porosity prediction: a neural network approach." IEEE Transactions on Geoscience and Remote Sensing 33, no. 4 (1995): 971-980.

7. Garson, David G. "Interpreting neural network connection weights." AI Expert (1991): 47-51.

8. Milne, Linda. "Feature selection using neural networks with contribution measures." In AI-CONFERENCE-, pp. 571-571. World Scientific Publishing, 1995.

9. Zhang, Wei, Gaoliang Peng, and Chuanhao Li. "Bearings fault diagnosis based on convolutional neural networks with 2-D representation of vibration signals as input." In MATEC web of conferences, vol. 95, p. 13001. EDP Sciences, 2017.

10. Bappy, Jawadul H., and Amit K. Roy-Chowdhury. "CNN based region proposals for efficient object detection." In 2016 IEEE International Conference on Image Processing (ICIP), pp. 3658-3662. IEEE, 2016.

11. Aytar, Yusuf, Carl Vondrick, and Antonio Torralba. "Soundnet: Learning sound representations from unlabeled video." arXiv preprint arXiv:1610.09001 (2016).

12. Oord, Aaron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. "Wavenet: A generative model for raw audio." arXiv preprint arXiv:1609.03499 (2016).

13. Kiranyaz, Serkan, Turker Ince, Ridha Hamila, and Moncef Gabbouj. "Convolutional neural networks for patient-specific ECG classification." In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2608-2611. IEEE, 2015.

14. van't Veer, Anna. "Effortless morality: Cognitive and affective processes in deception and its detection." (2016).

15. Tang, Wensi, Guodong Long, Lu Liu, Tianyi Zhou, Jing Jiang, and Michael Blumenstein. "Rethinking 1d-cnn for time series classification: A stronger baseline." arXiv preprint arXiv:2002.10061 (2020).