Detecting Subject Belief using Bidirectional Neural Networks with Genetic Algorithm Feature Selection and Auto-associative Transfer Learning

Emma Liu
Research School of Computer Science,
Australian National University,
Canberra Australia
u5368028@anu.edu.au

Abstract. Studies suggest that humans have the ability to unconsciously detect whether another person is telling the truth or not. This can be detected in physiological responses such as pupillary dilation and skin temperature. Zhu et al. [1] explored this connection by measuring the pupillary responses and analysing the data using a neural network. They were able to achieve 58.3% accuracy in distinguishing whether someone was presenting information they believed was true or misleading. In this paper, a similar study was conducted using bidirectional neural networks (BDNNs), augmented by genetic algorithm (GA) feature selection and transfer learning from auto-associative networks (AANs). While the BDNN trained on pupillary dilation data was able to achieve results above chance level, it was not able to match or exceed an accuracy of 58.3%, suggesting that the bidirectional functionality may not improve model performance. However, results could be improved by using a combination of physiological features including skin temperature, galvanic skin response and blood volume pulse. Using this dataset, a base BDNN was able to achieve an accuracy of 61.1%. While not improving on the base BDNN performance, BDNNs using GA feature selection and BDNNs using AAN transfer learning were able to produce accuracies greater than chance level, achieving accuracies of 56.5% and 57.7%, respectively.

Keywords: Neural networks, Bidirectional neural network, Subject belief, Doubt, Pupillary dilation, Blood volume pulse, Galvanic skin response, Skin temperature, Feature selection, Genetic Algorithms, Auto-associative networks, Transfer learning

1 Introduction

While humans have the tendency to assume that people are generally honest [2], lies and mistruths are prevalent within society today. Technology has provided a platform for uninformed, unsubstantiated or purposefully misleading perspectives to be publicised and communicated to large audiences. Some lies have limited negative impact, but there are some that can have significant repercussions on a person's personal, professional, and public life [3]. The ability to distinguish whether someone is being deceitful is important in navigating this complex environment. However, studies have shown that on average, a person can only accurately identify 54% of statements as true or false [4]. People have an even greater difficulty at identifying deception, with 47% of deceitful statements being correctly identified. There is a clear demand for technology that can assist people in distinguishing honesty from falsehood.

There is evidence that suggests that humans' intuitive responses and biometric information can be used to improve the accuracy when distinguishing deceit from truthfulness. Experiments performed by Albrechtsen, Meissner and Susa [5] suggest that people who formed intuitive assessments have a significantly higher accuracy at detecting deceit compared to people using cognitive processing. Van 't Veer et al. [6] found that when observers view a person who is lying, they could detect a decrease in skin temperature, regardless of whether the observer was forewarned about the possible deceit or not. Both these studies suggests that a person can unconsciously detect whether they are viewing truth or deception.

Zhu et al. [1] also explored the relationship between observing deceit and the observer's biometric responses by measuring the observers' pupillary responses. To produce their dataset, they recorded four videos, each containing an individual presenting on a certain topic. In two of the videos, the presenters were told that the material they were presenting was "a bit bogus", manipulating the presenter into thinking they were presenting misinformation. Each participant watched the two of the videos while their pupillary size was measured. The study identified that there was a statistically significant difference in the pupil dilation response of the observer when observing a presenter who doubted their information as opposed to a presenter who believed the information they were presenting. When the pupillary responses were processed through a neural network to detect the presenter's belief, they achieved an accuracy of 58.3% which was much higher than the observer's conscious judgement of 50% accuracy.

This study aims to assess whether it is possible to achieve similar results to Zhu et al.'s accuracy of 58.3% for distinguishing presenter's subject belief by processing an observer's biometric data using a bidirectional neural network (BDNN). Like most neural networks, BDNNs are able take an input pattern to produce output data. However, they are trained to also be able to take the output data to predict the input data, mimic the human ability to draw associations between two corresponding concepts from either direction [7]. The models used in this study are based on Nejad and Gedeon's [7] paper 'BiDirectional Neural Networks and Class Prototypes'.

To test whether BDNN results could be improved, GA feature selection and transfer learning from several types of AANs were implemented to the base BDNN models. GA feature selection uses a genetic algorithm to choose only input features that contribution to the accuracy of the model to be used. This would improve the training convergence and may improve the performance of the model due to the removal of non-informative features. AANs were trained on the dataset to implement transfer learning as the first half of the AAN layers compressed the input features, removing input data that provided little contribution to the output of the model. This would also improve training convergence, as well as focus the BDNN model on higher level features from the data, which may help generalisation. Standard feed forward AANs were used, as well as AANs with shared weights and bidirectional AANs, as outlined by Gedeon, Catalan and Jin [8]. The symmetry of the weights resulting from these two AAN models were hypothesized to assist with the bidirectional training of the BDNN.

The dataset used to train the BDNN was taken from Zhu et al.'s [9] study on Deceit Detection. This dataset was produced using a similar method as the Zhu et al. [1] study with the exception that it also includes data on the observers' blood volume pulse (BVP), galvanic skin response (GSR) and skin temperature. These datasets were also incorporated into the BDNN's training to assess whether these biometrics could improve the model's ability to distinguish the presenter's subject belief.

2 Method

2.1 BDNN architecture

Four BDNN architectures were used in this study. The BDNNs were based on multilayer perceptron networks, two models with one hidden layer and two models with two hidden layers. Each layer of the models except for the output layer used an activation function, either the ReLU function or the sigmoid function.

To achieve the bidirectional functionality of the BDNN, two separate workflows were specified for the forward function of the model (Table 1.). This ensures that when the model was set to reverse, the data passed through each layer in the reverse order. For this to work, each time the direction that data was passed through the model was changed, the weights for each layer had to be transposed. Conceptually, this ensured that the weights used were the same for both directions between the same two layers. Additionally, the same bias terms were used between different layers of the model, depending on the direction setting of the model. For example, in the TwoLayerBDNN (Table 1.), the bias term b_h is used between the hidden layer and output layer during the forward direction and between the hidden layer and the input layer during the reverse direction.

Table 1. Forward functions of BDNN models. In the following equations, x is the input data, y is the target data, x^* is the input prediction, y^* is the target prediction, W_j are the weights, b_j are the bias terms and σ represents the sigmoid function.

	TwoLayerBDNN	TwoLayerBDNNSig	ThreeLayerBDNN	ThreeLayerBDNNRelu
Forward direction	$h = ReLU(W_1x + b_h)$ $y^* = (W_2h + b_o)$	$h = \sigma(W_1x + b_h)$ $y^* = (W_2h + b_o)$	$h_1 = ReLU(W_1x + b_{h_1})$ $h_2 = \sigma(W_2h_1 + b_{h_2})$ $y^* = (W_3h_2 + b_o)$	$h_1 = ReLU(W_1x + b_{h_1})$ $h_2 = ReLU(W_2h_1 + b_{h_2})$ $y^* = (W_3h_2 + b_o)$
Reverse direction	$h = ReLU(W_2^Ty + b_h)$ $x^* = (W_1^Th + b_i)$	$h = \sigma(W_2^Ty + b_h)$ $x^* = (W_1^Th + b_i)$	$h_2 = \sigma(W_3^Ty + b_{h_2})$ $h_1 = ReLU(W_2^Th_2 + b_{h_1})$ $x^* = (W_1^Th_1 + b_i)$	$h_2 = ReLU(W_3^Ty + b_{h_2})$ $h_1 = ReLU(W_2^Th_2 + b_{h_1})$ $x^* = (W_1^Th_1 + b_i)$

2.2 Dataset description and normalisation

The dataset was generated from 23 participants, with each participant responding to 16 different video clips. This resulted in a total of 368 individual patterns in the dataset. The input data for each pattern included 34 BVP features, 23 GSR features, 23 skin temperature features and 39 pupillary dilation (PD) features. The target data consisted of a single binary feature, where 1 indicated that the presenter's belief was not manipulated 0 indicated that presenter's belief was manipulated.

Since the mean values of certain features corresponding to the participants' BVP were significantly higher than the mean values of other features (Fig. 1.), it was important to normalise the data to ensure those features did not have more influence during training compared to other features. The input features were normalised across all data patterns for each participant, as this reduced the effect of individual bias [1] and improved the accuracies being achieved.

The split between the manipulated belief class to the non-manipulated belief class was 47.6% to 52.4%, respectively. As this is a relatively even split, it was decided that accuracy of the model prediction would be a reasonable method to assess model performance.

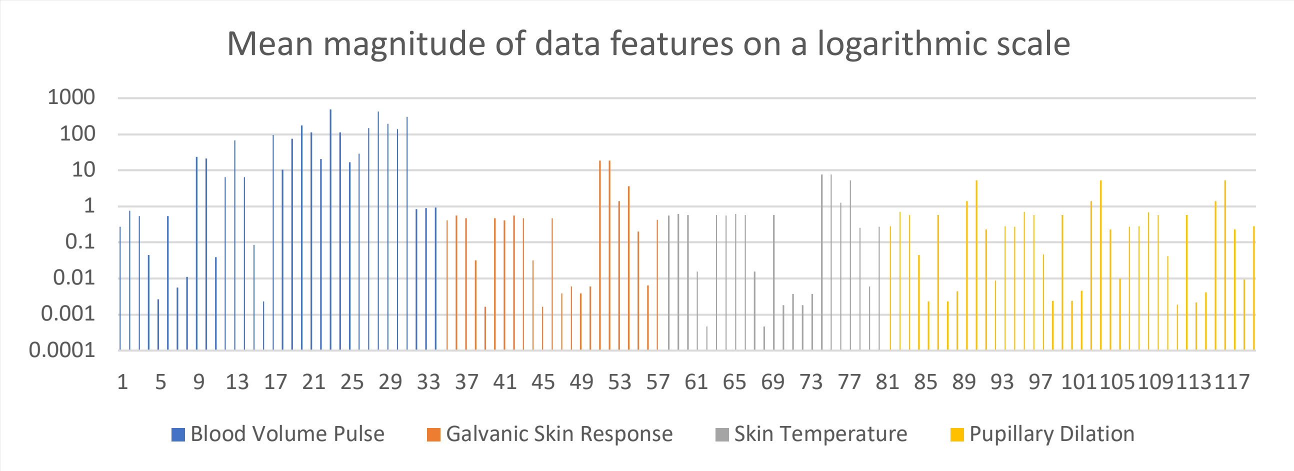

Fig. 1. Magnitude of means of data features displayed on a logarithmic scale

2.3 Training settings

BDNNs can be trained to predict the input data from target data as either associative memories or as cluster centre finders. Initial testing found that cluster centre training could achieve higher accuracies then associative memory. As a result, cluster centre training was used throughout this study. All training was conducted with the Adam optimiser and the backpropagation technique for both the forward and reverse directions. Cross entropy loss was used for the forward direction and mean squared loss was used for the backward direction.

To generate the final accuracies of the trained models, cross-validation was used. As it is important that all data patterns from the one participant remain in the same dataset either used for training or testing, each cross-validation trial used two participants' data for testing, with the remaining twenty-one participants used for training. This was similar to the leave-one-participant-out method used by Zhu et al. [1] but slightly less computationally expensive.

To determine when to switch between forward and reverse training, each model was initially trained in the forward direction for a specified number of epochs (the maximum number of epochs per direction). At the end of the specified number of epochs, the average loss per target node over the latest epoch, designated as the error tolerance, was recorded and the direction of training was switched. In subsequent epochs, if the number of epochs in the same direction exceeded the maximum number of epochs or if the average loss per target node went below the error tolerance, the training direction was switched, and the error tolerance was updated if necessary.

2.4 Hyperparameter tuning

To establish a baseline performance from a standard neural network on the dataset for this model, hyperparameter tuning was performed on the BDNN models with the reverse function disabled. Hyperparameters consider were the model architecture, the number of hidden neurons, the number of epochs, the learning rate and weight decay. 287 hyperparameter settings were investigated.

The hyperparameters considered when tuning the BDNN models were the model architecture, the number of hidden neurons, the number of epochs, the maximum number of epochs before training switched directions and the learning rates and weight decay for both the forward and reverse directions. In total, 751 different hyperparameter settings were tested during hyperparameter tuning. The hyperparameter settings for the best basic BDNNs can be found in Table 2. in the appendix of this report.

For the majority of the hyperparameter tuning, cross-validation was not used. Instead, a random set of two participants' data was selected as the validation set, reducing the time required for the hyperparameter tuning.

2.5 Feature selection using GA

In the GA used to select input features for the BDNN, each feature was represented by a binary number, where 1 represents including the feature, and 0 represents excluding the feature. Crossover was implemented using two-point crossover and mutation occurred through randomised bit flips. The fitness of the representation was found by calculating the accuracy from a BDNN model trained on the input features identified by the GA representation. Cross-validation was not used as it would slow the GA significantly. The BDNN model selected for the fitness function was the best performing two layered BDNN (Table 2.), as it would be faster to train then a three-layered model. Selection was implemented using tournament selection. GA setting values can be found in the appendix (Table 3).

Once the optimal input features were identified, a second round of hyperparameter tuning was implement on the BDNN using the selected input features. 596 different hyperparameter settings were tested. (See Table 4. for the hyperparameter settings and accuracies of the BDNNs with GA features selection.)

2.6 Transfer learning using auto-associative networks

The AANs used in this experiment had an architecture with three hidden layers and 119 input and 119 output features. The first and second hidden layers each had fewer hidden neurons then the previous layer, ensuring that the input information was compressed. The third hidden layer had the same number of neurons as the first hidden layer, resulting in a symmetrical layer structure (Fig. 2.). The ReLU function was used as the activation function for all layers.

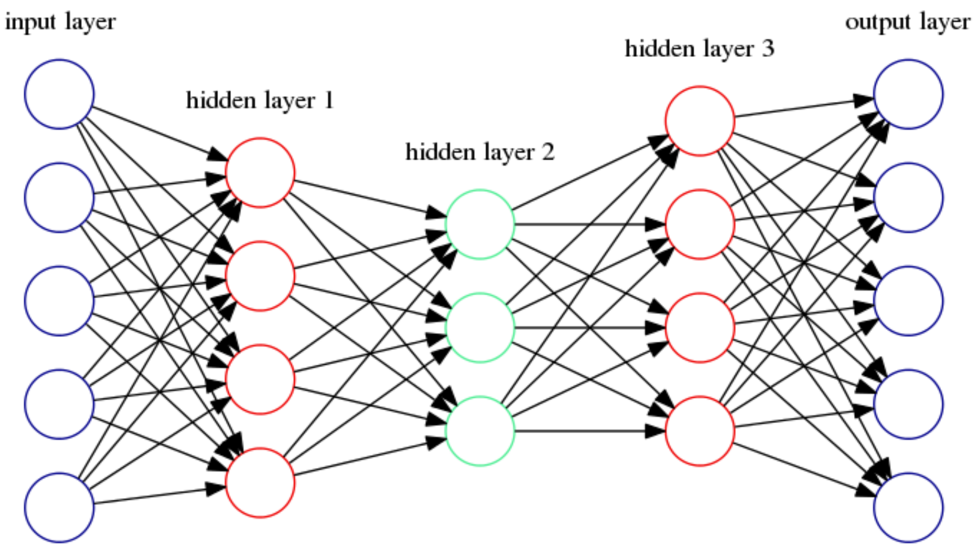

Fig. 2. AAN structure. The number of hidden neurons in each layer of the AAN models is greater than what is shown in the figure.

For AAN training with shared weights, after each time the model was optimised, the weights and biases on each pair of symmetrical layers were updated by the average weight and average bias of the two layers, resulting in identical weight and bias values on each pair of symmetrical layers. For the AAN with bidirectional training, each time the model was optimised, the weights and biases of each pair of symmetrical layers were swapped, resulting in bidirectional training over the weights and biases. Each AAN was had hyperparameter tuning, with a total of 689 settings tested for the standard AAN, 434 settings tested for the AAN with shared weights and 534 settings tested for the AAN with bidirectional training. (See Table 5. for the hyperparameter settings for the best AAN models.)

The BDNN models used with transfer learning was the ThreeLayerBDNN and the ThreeLayerBDNNReLU as those models had enough layers to implement two pretrained layers from the best performing AANs. Hyperparameter tuning was performed on the BDNN with either the first layer or the first two layers copied from the AAN models. During BDNN training, the parameters of the transferred layers were either fixed, optimised at the same learning rate as the rest of the model or finetuned, where the optimisation learning rate was significantly smaller than the rest of the model. A total of 331 settings were tested on the BDNN with the regular AAN transfer learning, 682 settings were tested for the BDNN with bidirectional AAN transfer learning, and 630 settings for the BDNN with shared weights AAN transfer learning. (See Table 6. for the hyperparameter settings and accuracies for the BDNNs with transfer learning.)

3 Results and discussion

3.1 Results using only pupillary dilation data

The best result achieved by the BDNN when trained on only PD data was an accuracy of 56.8%. This model had a TwoLayerBDNN architecture. While above a chance level of performance, this result does not match the accuracy attained by Zhu et al. [1] of 58.3%. The BDNN's performance only slightly improved on the accuracy of 56.0% which was achieved by the neural network using the same BDNN architecture but with the reverse function disabled. This suggests that BDNNs may not improve performance of distinguishing a presenter's subject belief using the PD of the observer.

From Fig. 3., it is evident that during the beginning of the training, the training accuracies are increasing as expected, once training is reversed at epoch 50, the training accuracy flattens out. In the test accuracy, the accuracy does not appear to significantly improve from the start of training, but the test accuracies remain mostly static after reverse training is introduced. This suggests that introducing a secondary goal to map cluster centres reduces the model's ability to improve its accuracies in the forward direction.

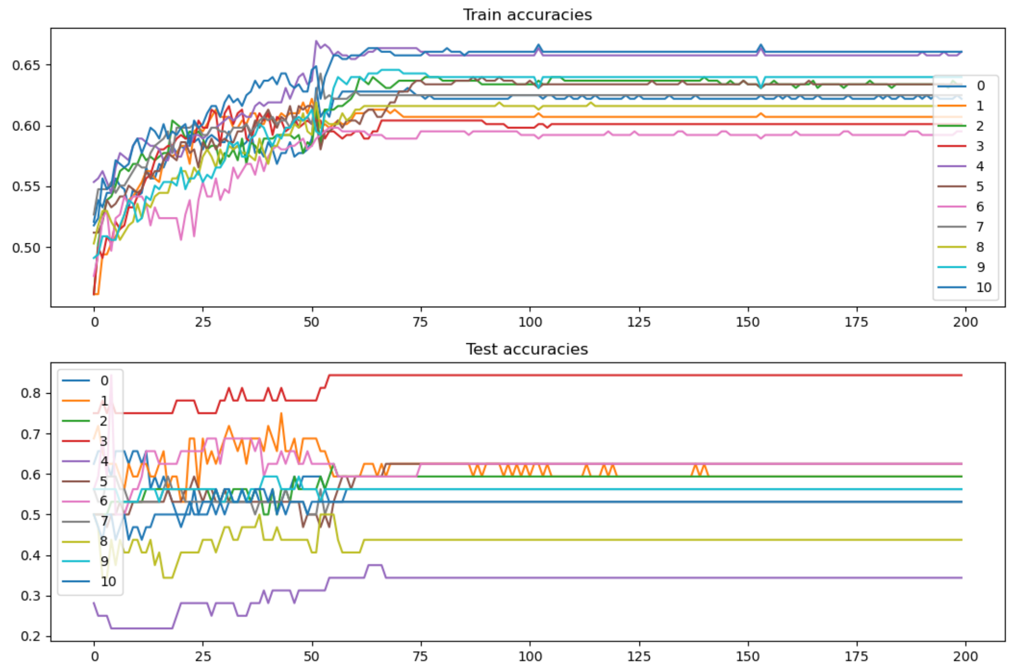

Fig. 3. Accuracies over number of epochs for the best BDNN model for PD data. Each line represents one cross-validation trial.

3.2 Best overall performance

The model that achieved the best overall accuracy had a ThreeLayerBDNN architecture. This model attained an accuracy of 61.1%, above both the result achieved by the original study's neural network and that of an observer's conscious judgement of 50% [1]. The non-directional neural network with the best accuracy also was trained using all features within the input data and achieved an accuracy of 58.0%.

From these results, it appears that using a range of biometric data rather than only one type of biometric data improves the performance of the model. This makes sense, as the model can use the multiple types of features from the dataset and the interrelation between those features to determine the observer's unconscious assessment of the presenter to form a result. This process also replicates how humans forms their assessments on the honesty of others. When judging whether others are speaking truthfully or with deceit, humans use a wide range of indicators rather than just a singular feature.

3.3 BDNN performance vs non-BDNN performance

There is only a small difference between the accuracies achieved by the bidirectional models and the non-bidirectional model. As with the models trained on PD information only, this indicates that the bidirectional functionality produces little improvement in producing an accurate result.

Like with the PD training, once the model started training in the reverse directions, the accuracies remained relatively stable for the rest of the training. This was observed across most of the training performed using a BDNN architecture and the reinforces the perspective that the bidirectional functionality provides little to no improvement in accurately identifying subject belief.

When comparing the testing accuracies of each cross-validation trial for both the best BDNN model (Fig. 4.) and the best non-BDNN model (Fig. 5.), it is evident that the accuracy of the results depend heavily on the participants within the test sets as each test set contains the data patterns measured from two participants in the original study. Both BDNN and non-BDNN results show drastically higher training accuracies compared to the test accuracies. These observations suggest that the specific biometric feature patterns that can be used to attain an accurate result for some people may not carry across for other people and that the models are overfitting to the training data.

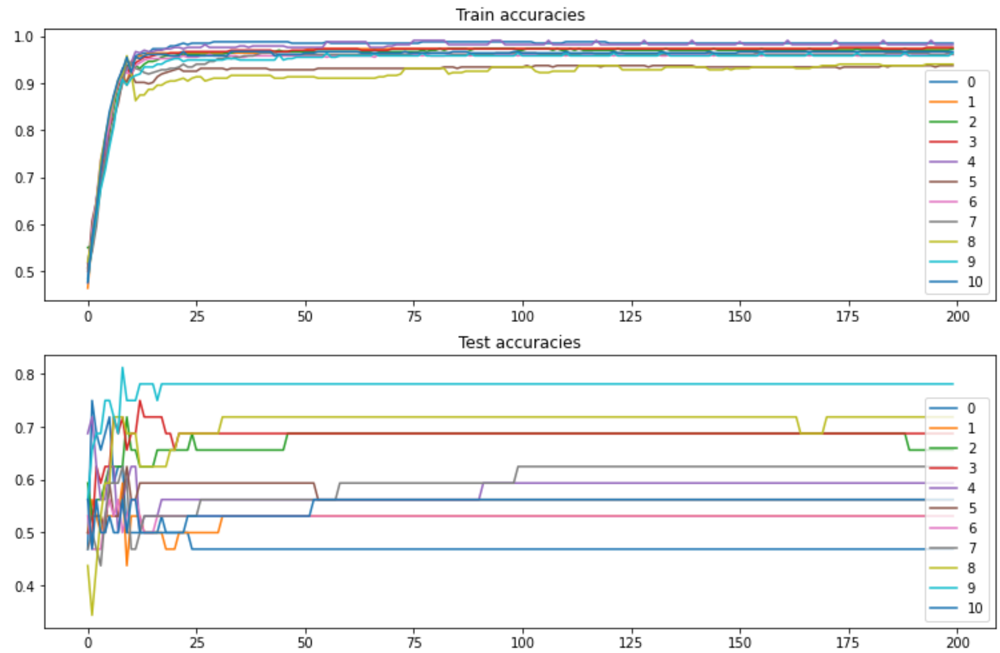

Fig. 4. Test accuracies over epochs for best BDNN model trained on all data

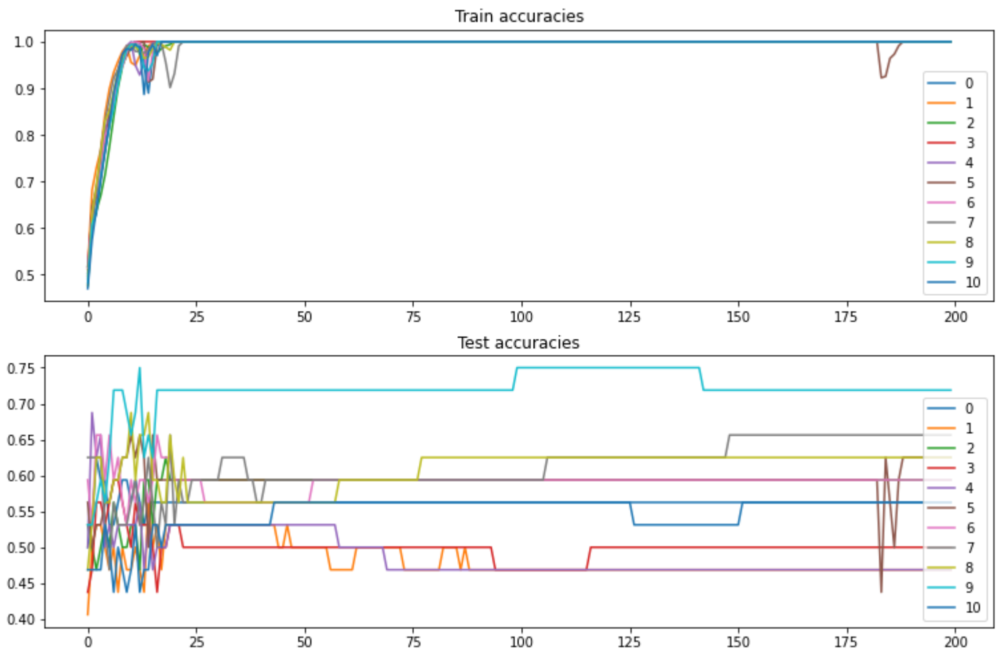

Fig. 5. Test accuracies over epochs for best non-BDNN model trained on all data

3.4 GA performance

The feature selection representation with the highest fitness had a fitness level of 81.3%. However, after hyperparameter tuning was performed on the models using only the selected features, the highest accuracy attained with cross-validation was 56.5%, which did not improve on the regular BDNN performance.

This may be due to limitations in the implementation of the GA. As shown in Fig. 6., the GA did not converge towards an optimal solution. Instead, performance of each generation was unstable. This could be caused by the limited area that the algorithm could explore in the feature space. Only a relatively small number of individuals and generations were used when running the GA, due to the amount of time fitness calculations took for each individual. Another reason why the GA did not converge may be due to the inconsistent accuracies produced by a BDNN, resulting with the fitness of a representation being unpredictable. During training, it was noticed that the same BDNN model can produce varying level of accuracies, which shows that BDNN performance has a high sensitivity to either the training and testing data sets, or the initialisation of the model parameters.

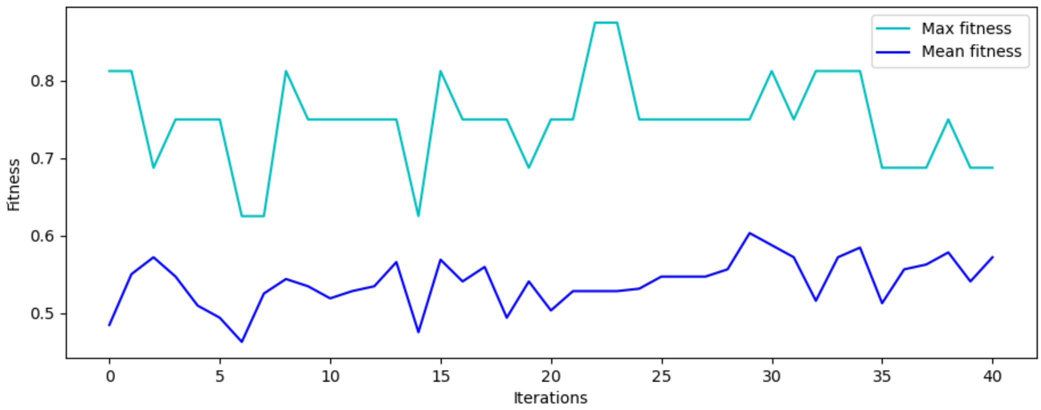

Fig. 6. Fitness values across GA iterations from one of the GA runs.

3.5 Transfer learning performance

The best performance achieved by a BDNN using transfer learning was an accuracy of 57.7%. This was achieved by three different transfer learning models: fixed parameters from a bidirectional AAN, fixed parameters from a regular AAN and finetuned parameters from a regular AAN. As demonstrated in Fig. 7., results suggest that fixed parameters perform best for BDNN transfer learning. A reason behind this may be the training restriction that fixed weights causes, which prevents the BDNN model from overfitting to the training data.

The regular AAN appeared to perform best in terms of accuracy out of the three types of AANs. However, further investigation is needed, focusing on the performance of the BDNNs in the reverse direction, before the best transfer learning model for BDNNs is established, as the bidirectional AANs and the shared weights BDNN were hypothesized to improve the reverse performance of BDNNs due to the symmetry of their weights.

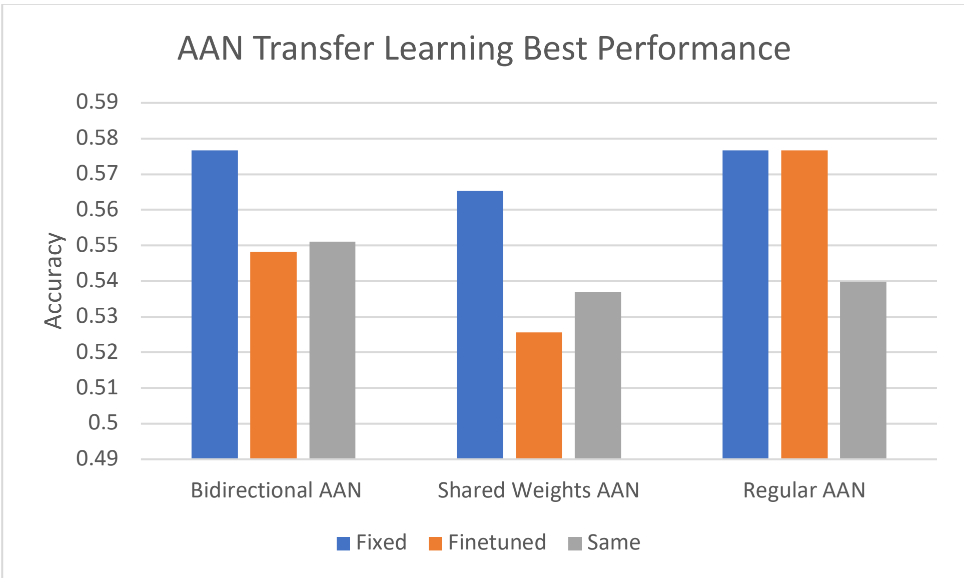

Fig. 7. Best accuracies achieved by BDNNs using transfer learning.

While the results did not improve on the accuracy achieved by the base BDNN, transfer learning for BDNNs still shows potential as a usable technique, as the best accuracy achieved is not significantly below that of the base BDNN. Further hyperparameter tuning may achieve results similar or greater than the base BDNN.

3.6 Limitations and improvements

Due to the size of the dataset provide, and the limited number of participants used to generate the data, this dataset may not capture a wide range of the unconscious response to a presenter's subject belief. As a result, the models trained on this data may have difficulty generalising the patterns that identify whether the presenter has a manipulated or unmanipulated belief and may not produce very accurate results if used on the general populace. One technique that may mitigate this issue is to reverse the training direction after training on one data pattern instead of after a certain number of epochs. This will increase the noise during training and may reduce the model's tendency to overfit.

There was difficulty in replicating model results, even when identical training settings were used. This suggests that model performance high dependence on the initialisation of weights and the order in which patterns were passed through the model during training, as these factors were the only randomised factors during model training.

As a majority of the hyperparameter tuning was not performed with cross-validation, this may have limited the search for the overall best hyperparameters. This was evident as the best hyperparameter tuning results achieved accuracies of 81.3% for the base BDNNs, the BDNNs using GA feature selection and the BDNNs using transfer learning. However, when cross-validation was implemented on a subset of the hyperparameter settings achieving the highest hyperparameter tuning results, the settings initially achieving 81.3% did not necessarily produce the best cross-validation accuracy. Hyperparameter tuning can be improved by evaluating hyperparameter settings using the cross-validation accuracy, though this will drastically increase the time needed for hyperparameter tuning.

4 Conclusion and future work

While BDNNs have some ability in distinguishing manipulated from non-manipulated subject belief using an observer's pupillary response, they were not able to match the accuracy achieved by Zhu et al.'s [1] neural network which was trained in only one direction. When trained with additional biometric information including skin temperature, GSR and BVP, the BDNN's accuracy could exceed Zhu et al.'s [1] result and achieve a 61.0% accuracy. This result also exceeded the baseline accuracy produced in this study of 58.0% from a standard neural network. However, further examination is required to determine whether the bidirectional functionality of the neural network improves the overall accuracy of the model.

This study could not utilise the full benefits of GA feature selection due to non-convergence of the GA during implementation. Results achieved in this study suggest that GA feature selection does not improve the accuracy of a BDNN. To fully explore this area, a greater population size and number of generations should be used in the GA. A larger dataset may also improve the stability of the GA fitness function.

The best performance achieved by a BDNN using AAN transfer learning was an accuracy of 57.7%. Results suggest that fixing the parameters of transferred layers during BDNN produces the best performance, as this may reduce the amount of overfitting in the model. The regular AAN appeared to perform the best out of the three types of AANs used in transfer learning in terms of accuracies. However, to determine whether transfer learning improved the overall performance of the BDNN model, additional investigation on the performance of the BDNN in the reverse direction, such as error metrics, should be taken.

An area that could be investigated regarding the dataset include the impact of the presenter on the accuracies on the finding. Zhu et al. [1] found in their study that the accuracy from one of their presenters was 60% higher than the other presenter. A similar effect may have happened with this dataset, which provide a greater insight into the overall accuracies of this study's models.

References

1. Zhu, X., Qin, Z., Gedeon, T., Jones, R., Hossain, M.Z., Caldwell, S.: Detecting the Doubt Effect and Subjective Beliefs Using Neural Networks and Observers' Pupillary Responses. In: International Conference on Neural Information Processing, Springer, Cham, pp. 610-621 (2018)
2. Levine, T.R.: Truth-Default Theory (TDT): A Theory of Human Deception and Deception Detection. Journal of Language and Social Psychology. vol. 33, no. 4, pp. 378–392 (2014)
3. ten Brinke, L., Stimson, D. Carney, D.R.: Some Evidence for Unconscious Lie Detection. Psychological Science. vol. 25, no. 5, pp. 1098–1105 (2014)
4. Bond, C.F., DePaulo, B.M.: Accuracy of deception judgments. Personality and Social Psychology Review. vol. 10, no. 3, pp. 214-234 (2006)
5. Albrechtsen, J.S., Meissner, C.A., Susa, K.J.: Can intuition improve deception detection performance?. Journal of Experimental Social Psychology. vol. 45, no. 4, pp. 1052-1055 (2009)
6. van 't Veer, A.E., Gallucci, M., Stel, M., van Beest, I.: Unconscious deception detection measured by finger skin temperature and indirect veracity judgments-results of a registered report. Frontiers in psychology. vol. 6, pp. 672-672 (2015)
7. Nejad, A.F., Gedeon, T.D.:, Bidirectional neural networks and class prototypes. In: ICNN'95 - International Conference on Neural Networks. vol. 3, pp. 1322-1327 (1995)
8. Gedeon, T.D., Catalan, J.A., Jin, J.: Image compression using shared weights and bidirectional networks. In: Proceedings 2nd International ICSC Symposium on Soft Computing (SOCO'97), pp. 374-381 (1997)
9. Zhu, X., Gedeon, T., Caldwell, S., Jones, R. Gu, X.: Deceit Detection: Identification of Presenter's Subjective Doubt Using Affective Observation Neural Network Analysis. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3174-3181. Toronto, ON, Canada (2020)

Appendix

Table 2. Hyperparameter settings and accuracy of basic BDNNs. Note that the following models had a batch size of 1 and weight decay of 0

	Model Architecture	Training data	No. of hidden neurons (layer 1)	No. of hidden neurons (layer 2)	No. of epochs	Max no. of epochs per direction	Learning rate (forward)	Learning rate (reverse)	Accuracy
Best BDNN (PD data)	TwoLayer BDNN	PD only	72	-	200	50	0.001	0.015	56.80%
Best non-BDNN (PD data)	TwoLayer BDNN	PD only	56	-	200	60	0.002	0.01	56.00%
Best BDNN (overall)	ThreeLayer BDNN	All data	24	72	200	10	0.0025	0.02	61.10%
Best non-BDNN	TwoLayer BDNN	All data	56	-	200	60	0.002	0.01	58.00%

Table 3. Training settings for the GA

Representation type	Representation length	Cross-over probability	Mutation probability	Population size	Number of generations	Number of GA runs
List of binary numbers	119	0.8	0.2	20 - 50	20 - 60	5

Table 4. Hyperparameter settings and accuracy of BDNNs with GA feature selection.

	Model Architecture	Batch size	No. of hidden neurons (layer 1)	No. of hidden neurons (layer 2)	No. of epochs	Max no. of epochs per direction	Learning rate (forward)	Learning rate (reverse)	Weight decay (forward)	Weight decay (reverse)	Accuracy
Best BDNN w/ GA features selection	ThreeLayer BDNNRelu	50	32	32	1000	75	0.00543	0.02711	0.01985	0.06691	56.80%

Table 5. Hyperparameter settings of AANs.

	Batch size	No. of hidden neurons (layer 1)	No. of hidden neurons (layer 2)	No. of epochs	Learning rate	Weight decay
Best regular AAN	40	110	50	1000	0.00459	0.00183
Best bidirectional AAN	50	110	50	200	0.00767	0.31031
Best shared weights AAN	50	100	90	1000	0.02227	0.00008

Table 6. Hyperparameter settings and accuracies of BDNNs with AAN transfer learning.

AAN model	Transfer layer optimisation	No. of transferred layers (TL)	Model Architecture	Batch size	No. of hidden neurons (layer 1)	No. of hidden neurons (layer 2)	No. of epochs	Max no. of epochs per direction	Learning rate (forward)	TL Learning rate (forward)	Learning rate (reverse)	TL Learning rate (reverse)	Weight decay (forward)	TL Weight decay (forward)	Weight decay (reverse)	TL Weight decay (reverse)	Accuracy
Regular	Fixed	1	ThreeLayer BDNN	50	110	32	150	20	0.02846	0	0.01381	0	0.08918	0	0.25386	0	57.7%
Regular	Finetuned	2	ThreeLayer BDNN	1	110	50	300	10	0.02757	0.00087	0.00772	0.00149	0.05703	0.28964	0.23222	0.12389	57.7%
Bidirectional	Fixed	1	ThreeLayer BDNN	50	110	16	300	20	0.01462	0	0.01047	0	0.02488	0	0.28690	0	57.7%
Shared weights	Fixed	2	ThreeLayer BDNN	10	100	90	150	20	0.00355	0	0.02988	0	0.18874	0	0.32544	0	56.5%
Bidirectional	Same	2	ThreeLayer BDNNRelu	1	110	50	500	20	0.02922	0.02922	0.02465	0.02465	0.14756	0.14756	0.46164	0.46164	55.1%
Bidirectional	Finetuned	2	ThreeLayer BDNN	30	110	50	150	10	0.04198	0.00044	0.01390	0.00378	0.05192	0.22658	0.10904	0.16568	54.8%
Regular	Same	2	ThreeLayer BDNNRelu	20	110	50	300	20	0.03182	0.03182	0.04717	0.04717	0.42656	0.42656	0.26991	0.26991	54.0%
Shared weights	Same	1	ThreeLayer BDNN	1	100	128	150	20	0.04892	0.04892	0.04151	0.04151	0.38240	0.38240	0.47678	0.47678	53.7%
Shared weights	Finetuned	1	ThreeLayer BDNN	10	100	128	300	20	0.03943	0.00389	0.01468	0.00340	0.32640	0.29265	0.25497	0.45703	52.6%