Predict the veracity of anger through multilayer perceptron (MLP) in different sizes/different number of layers and explain the results through analyzing their performances

Tong Cai
Research School of Computer Science
The Australian National University
Canberra ACT 2601
U6619441@anu.edu.au

Abstract: this paper issues a binary classification problem to predict the veracity of anger when observing participants' pupillary responses in watching a series of videos, as an extension of deep learning approach, multilayer perceptron (MLP) models in different sizes/different number of layers are trained to solve the classification problem, as well as analyzing the effect of size in MLP models on their performances. Pupillary responses are seen as a more valuable signal than verbal responses to predict whether the anger presented by the participants is real or posed. After training 9 MLP models in different size, in general, those MLP models with larger sizes or more layers performs better than smaller MLP models in predicting the veracity of anger, but some very larger MLP models present a poor performance.

Keywords: MLP, performance, veracity of anger

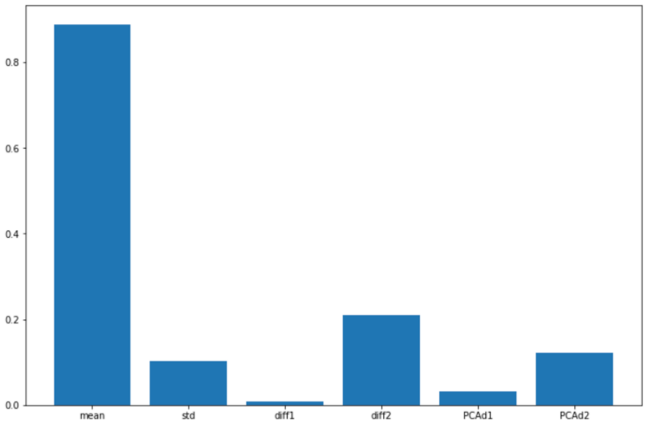
1 Introduction

This experiment collects the pupillary responses from 20 participants in watching 20 different videos to examine human's non-conscious and conscious ability to detect the veracity of anger[1].

1.1 Data inspection

The dataset used in this paper is Anger_v1.

Fig 1. visual results for the raw data (pupil movements)



There are 20 participants in this experiment, and each of them is expected to watch 20 videos.

The dataset contains 400 sets of data, each set of data records the pupillary responses like the mean value, standard deviation of pupil movements (Mean, Std, Diff1, Diff2, PCAd1, PCAd2) in two eyes, as well as the emotional response of each participant after watching this video (label). Among all 20 videos, a participant expressed real anger in watching 10 videos of them and expressed posed anger for the remaining 10 videos (where T1-T10, F1-F10 shows the number of videos)

1.2 Classification problem devised

After inspecting the raw data, I devised a binary classification problem to predict whether a participant is really angry or posed angry in watching different videos based on the statistical data of pupillary responses collected from all participants. This problem is worth solving because classification problem is a widely used in many areas, and for this experiment, it is helpful to analyze the effect of pupillary response in predicting human emotions.

1.3 Data preparation

To predict whether the anger of participants is genuine or posed, I dropped the index column and the column 'Video' which contains 20 different video numbers, then I encoded "Genuine" to 0 and "Posed" to 1 in the column "label". For the remaining columns of numerical values, I normalize those values to make them ranges from 0 to 1, this step is to avoid some numerical problems in training the MLP models.

After that, I randomly divided the preprocessed dataset into the training dataset (90%) and a testing dataset (10%), the training dataset is used to train parameters within the MLP models, while the testing dataset is used to examine the state and convergence of MLP models during the training process.

1.4 Multilayer perceptron

I designed a MLP model to solve the binary classification problems.

MLP models is a kind of feedforward neural network, which is generally composed of an input layer, hidden layers and an output layer.
The number of neurons on input layer is based on the number of features selected to predict the output, in this experiment, there are 6 features (Mean, Std, Diff1, Diff2, PCAd1, PCAd2), hence the input size is 6. As for the number of neurons on output layer, there are two kinds of output, which are 0 and 1 respectively, hence the output size is 2.
MLP models are suitable to handle classification problem in the experiment. Furthermore, they are flexible to learn the mapping from input to output, this flexibility makes them applied to many kinds of data including the dataset used in this experiment[2].
In order to better analyze MLP models' performances in solving such problems, I change the number of hidden layers and the number of neurons in each hidden layer to observe how different configurations affect model performance.

2 Method for MLP in different sizes / different number of layers

2.1 Implementation

The parameters for training MLP model are as follow:
Optimizer = Adam
dropout = 0.5;
learning rate = 1e – 2;
batch size = 1024;
epoch = 100
The learning rate are adjusted after each epoch using the One Cycle learning rate scheduler provided by pytorch. The pct_start is set to 0.3 and annealing strategy 'cosine', as these empirically produces best results. Within each epoch, the model is exposed to 32,768 samples. I discovered that fewer total samples would prevent the model from converging. Through multiple testing, I discovered that setting dropout to 0.5 is the most successful in precenting overfitting while balancing overall model performance.

Gaussian Error Linear Unit (GELU) is adopted as the activation function to make neuron output probabilistic, it is related to random regularization to be the expectation of adaptive dropout correction [3]. The effect of dropout is to increase the generalization ability of MLP model and reduce overfitting. It is also documented in the original paper that GELU helps with model convergence.
Both training loss and training accuracy, testing loss and testing accuracy is recorded for each epoch in all epochs.

2.2 change the network structure

MLP models in this setting have fixed input size and output size, hence the chosen method to compare model performance vs model size and depth is to adjust the number of layers and number of neurons within each layer.

To make a more comprehensive comparison on MLP models' performance, I have trained 9 different MLP models. The number after MLP is the number of parameters in that model.
The number of parameters in a MLP model also presents its size.
There are three MLP models with same number of hidden layers (two hidden layers), but the number of neurons on each hidden layer is different, the detailed parameters for each of the three MLP models are:
MLP_176: 10 neurons on the first hidden layer, 8 neurons on the second hidden layer
MLP_356: 12 neurons on the first hidden layer, 18 neurons on the second hidden layer
MLP_508: 14 neurons on the first hidden layer, 24 neurons on the second hidden layer
Through comparison of output after training those three MLP models, it would be convenient to observe how their performance changes according to the size changes.

Moreover, to observe how MLP models' performances related to their size and number of layers, there are other MLP models both in different size and with different number of layers.
The detailed parameters for each of the 6 MLP models are:
MLP_110 (1 hidden layer): 12 neurons on the hidden layer
MLP_228 (2 hidden layer): 10 neurons, 12 neurons

MLP_830 (3 hidden layer): 12 neurons, 24 neurons, 16 neurons
MLP_1678 (4 hidden layer): 12 neurons, 16 neurons, 32 neurons, 24 neurons
MLP_4298 (5 hidden layer): 12 neurons, 24 neurons, 36 neurons, 48 neurons, 24 neurons
MLP_6612 (6 hidden layer): 12 neurons, 24 neurons, 36 neurons, 48 neurons, 64 neurons, 6 neurons

3 Methodology of technique

3.1 Understanding of the technique

From my understanding, the technique is to adopt a method to generate rules, combining the goal of experiment to predict the authenticity of anger, the technique introduces an approach to predict the neural network's output by setting rules, and the specific model need to be built to fit the rules. After that, a decision tree model is trained to predict the authenticity of anger using the decision tree rules.

3.2 Decision tree

Decision tree is sensitive in solving classification problem, similar as a set of if-then structure, decision tree could do some feature-based probability distribution of the target requirement and represents the process of classifying the dataset based on the features.

Each node in the decision tree represents the features in the dataset, and the leaf node represents the output in the dataset, while the link between each nodes represents the divide rules (the probability to the next feature).[4]

Training a decision tree involves three steps, which are selecting the features, generating a tree model and pruning decision tree respectively.

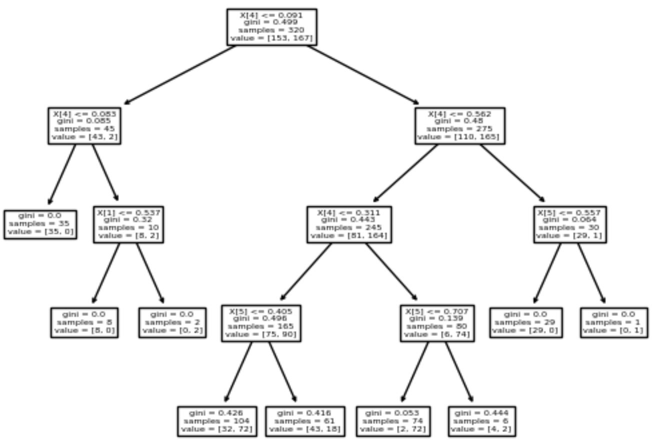
Information gain is a significant factor to select those features with greater classification ability. The decision tree model is built from the root node based on the selected features. After that, to avoid overfitting problem in a too complicated tree model, pruning provides help to solve the problem by pruning some nodes or subtree in the decision tree.

3.3 Implementation and result

Similar as training MLP models, after preprocessing the data and split training and testing dataset, I created a decision tree model with depth 4 to predict the veracity of anger.
The figure below shows both training accuracy and testing accuracy
Table 1. Training and testing accuracy of 3-different MLP models

	Training accuracy	Testing accuracy
Decision tree	83%	79%

Fig 2. decision tree with depth = 4



4 Results and discussion

4.1 MLP models with same number of layers but different size

First look at the training loss and testing loss

Fig 3. training loss during 100 epochs

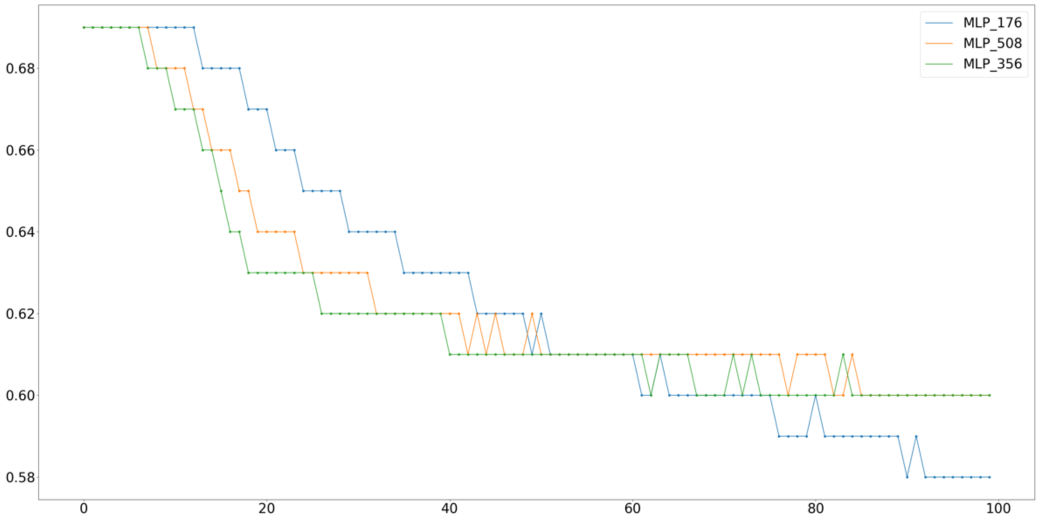
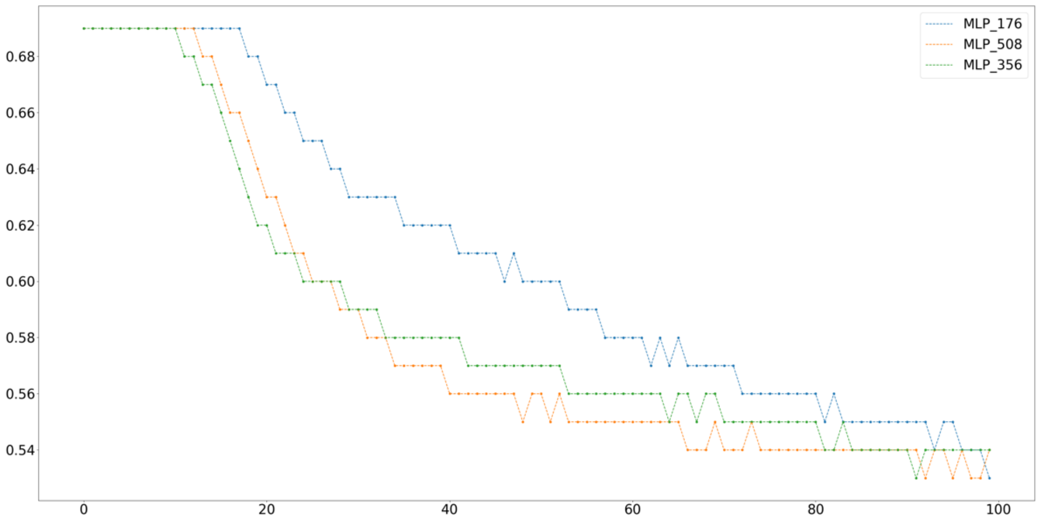


Fig 4. testing loss during 100 epochs



Observation:
the two graphs above shows both training loss and testing loss for three different MLP model with same number of hidden layers but different size. All three models' training and testing loss decrease throughout the training process; however, it appears that the smallest model (blue line) continue to decrease in training loss while the other two stagnated. While all three model eventually produced similar testing loss, the largest model (orange line) shows fastest convergence speed, while the smallest model is slower in converging. This is a surprising behavior for me, because I was expecting testing loss trend to be similar to training loss trend.

Below two graph shows training accuracy and testing accuracy.

Fig 5. training accuracy during 100 epochs

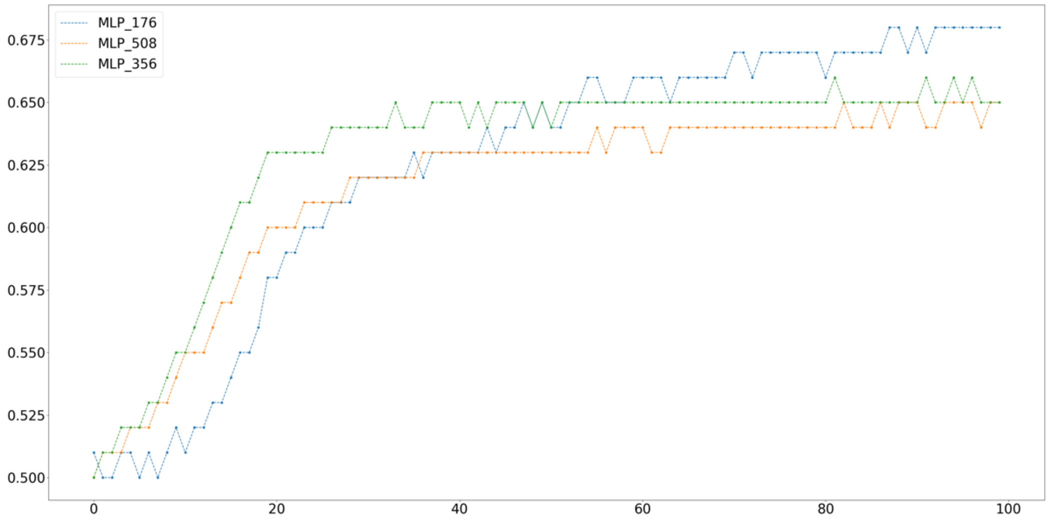


Fig 6. Testing accuracy during 100 epochs

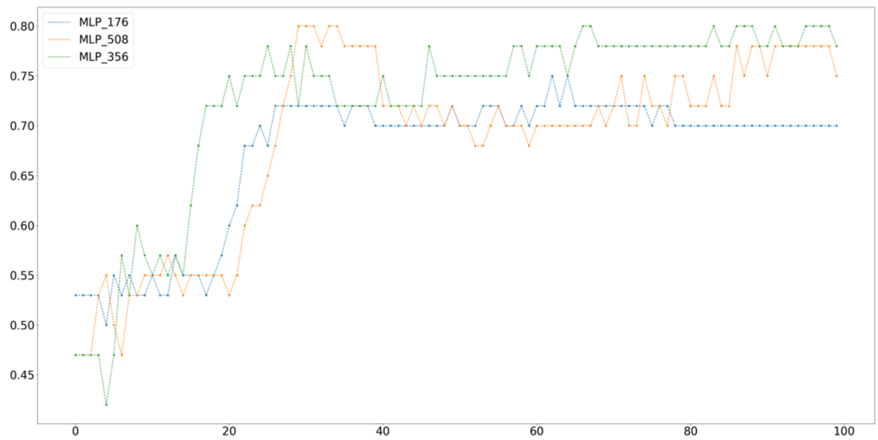


Table 2. Training and testing accuracy of 3-different MLP models

	Training accuracy	Testing accuracy
MLP_176	68%	70%
MLP_356	65%	78%
MLP_508	65%	75%

Observation:
From the two graphs above, for those three MLP models in different size, both training accuracy and testing accuracy increases during 100 epochs. However, the MLP model with largest size shows the largest increase in testing accuracy, while the model with smallest size shows the smallest increase. The two larger MLP models present a higher accuracy in predicting the veracity of anger. Having said that, the middle-sized model produced best classification result in testing dataset, contradicting findings when examining the cross-entropy loss. This might be caused by the middle-sized model does not as confident as others with probability, but generally leans toward the correct result. Also, the largest model is much more unstable when examining the testing accuracy compare its counterparts. The smallest model continued to suffer in testing accuracy, potentially showing that larger model can learn more knowledge and make better predictions.

4.2 MLP models with different size and different number of layers

Fig 7. training loss during 100 epochs

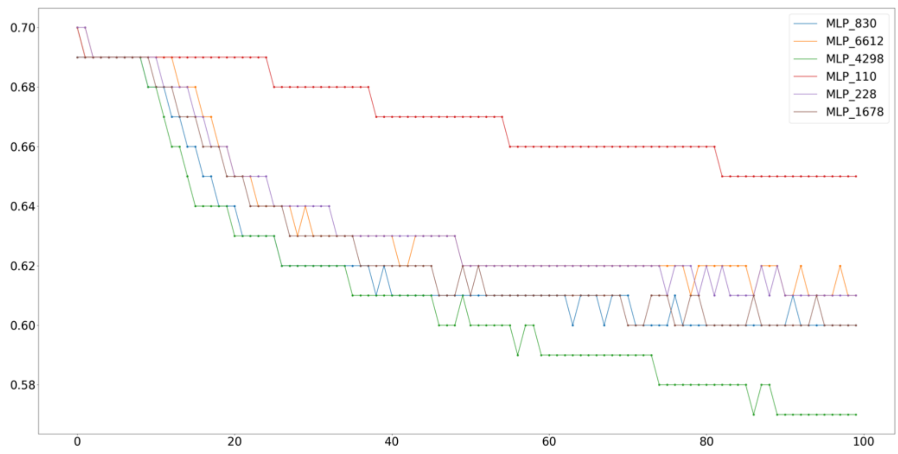
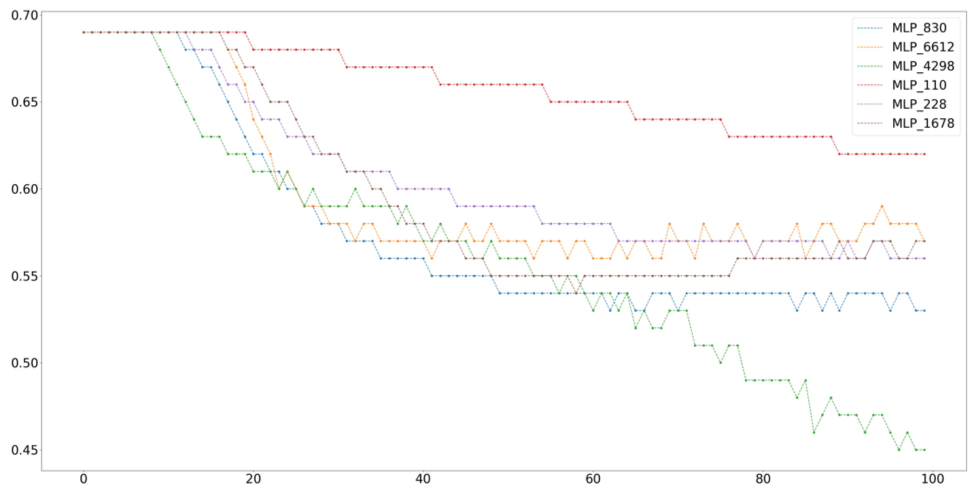


Fig 8. testing loss during 100 epochs



Observation:
All six models' training loss and testing loss continue to drop throughout the training process. Nonetheless, the MLP model with largest size and most hidden layers is not the one which shows the greatest decrease in loss. When evaluating training loss and testing loss, the largest model only ranked fifth. This indicates that larger model does not learn as much information as the smaller ones, potentially due to the limited dataset size. However, as expected, the smallest model (red line) had the worst performance in both training and testing, indicating that not enough information is captured due to its restricted size. The best performing model (MLP_4298 green line) is with size 4298 and 5 hidden layers. The model's training and testing loss continue to drop even after all other counterparts' have converged. MLP_4298 superseded all other models' performance by a significant margin.

Fig 9. training accuracy during 100 epochs

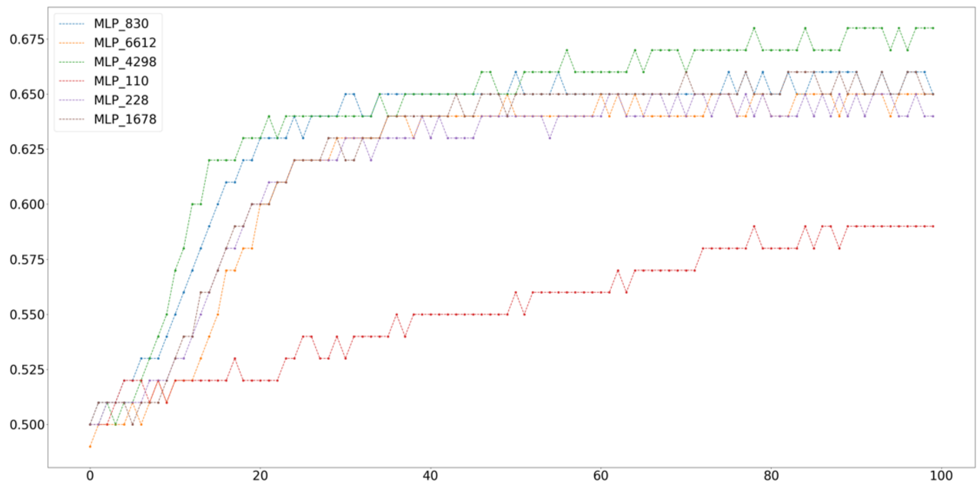


Fig 10. testing accuracy during 100 epochs

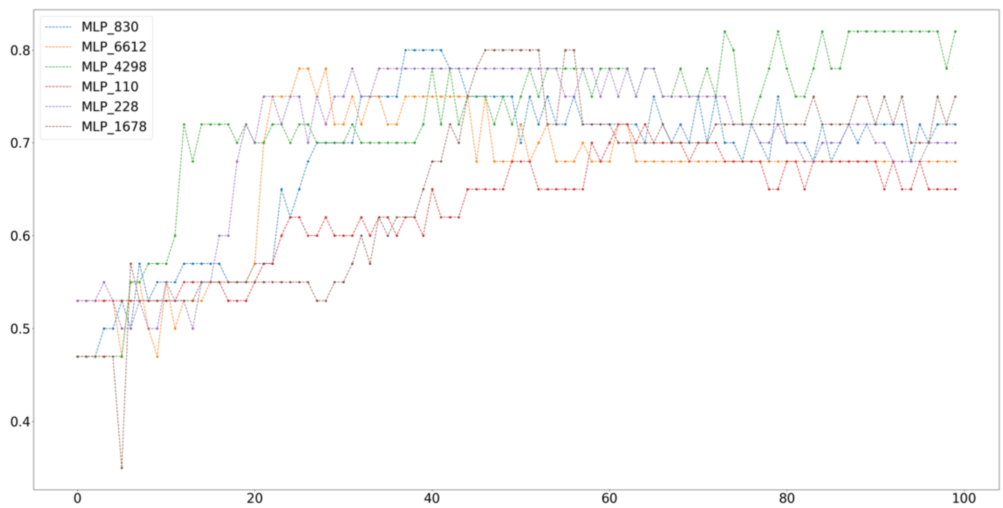


Table 3. Training and testing accuracy of 6-different MLP models

	Training accuracy	Testing accuracy
MLP_110	59%	65%
MLP_228	64%	70%
MLP_830	65%	72%
MLP_1678	65%	75%
MLP_4298	68%	82%
MLP_6612	65%	68%

Observation:

After training 100 epochs, all 6 models show improvement on their training accuracy and testing accuracy in predicting the veracity of anger.

The accuracy trend is similar to the cross-entropy loss trend, with MLP_4298 producing the best result in both training and testing, while the largest and smallest model suffers in accuracy score. It is worth pointing out that MLP_1678 (brown line) exhibited significant overfitting behavior when examining testing accuracy. This is indicative that an appropriately sized model not only can generate the best result, but it also has the potential to recover from or avoid overfitting issue. In general, when examining testing accuracy, the overall performance increases with the number of parameters, until it reached a peak and start to produce less accuracy results.

4.3 Comparison of MLP model and decision tree model

I choose the MLP model with the best performance to compare with decision model in solving the same binary classification problem based on the same dataset.

Table 4. Training and testing accuracy of MLP and decision tree

	Training accuracy	Testing accuracy
Decision tree	83%	79%
MLP_4298	68%	82%

Observing the form above, the testing accuracy of decision tree and MLP model are very close, but decision tree has a higher training accuracy than MLP model, showing overfitting issues.

4.4 Discussion

From the comparison of the performance in MLP models with different sizes, it is obvious that as the size of MLP model increases, its performance also increases until it reached a certain point. In general, deeper MLP is more flexible in fitting more complex input features. A MLP model contains more hidden layers has better nonlinear expression ability to learn more complex transformation. The benefits of adding more hidden layers or neurons can be summarized as follow. First, more parameters(greater size) allow the model to express more detailed nonlinear correlation. Second, increasing the number of layers allow improved layer-by-layer feature learning ability [5].

However, through comparing different model configurations, it is obvious that the increase in model performance through adding more parameters can reach a threshold, and after which will hinder the model's performance. In model comparison, MLP_4298 outperformed MLP_6612, demonstrating said behavior. This further point to the necessity of choosing the correct model configuration, as the appropriate model generates the best result and have less chance of overfitting.

Through examining the difference in training loss and testing loss, it appears that the MLP models are less prone to overfitting, and this is further demonstrated when evaluating with accuracy. The testing error tends to be lower than training error for MLP models, while decision tree exhibited overfitting even with depth 4. The overfitting issue will only be exacerbated when max depth is increased as training dataset become larger and more complex.

5 Conclusion and future work

In this experiment, in order to predict whether the participant is really angry or performed angry, I designed a binary classification problem. MLP model and decision tree model are utilized to compare performance. Building from findings in last paper, I extend the previous model using a deep learning approach by scaling the MLP models to observe how the performances changes based on different number of parameters and hidden layers.

After training 9 different MLP models and observing their performance on predicting the veracity of anger, the performance of MLP model could be improved by adding the number of neurons on each hidden layer and deepening the network structure (adding the number of hidden layers). However, it does not mean that the deeper MLP with larger size always perform the best, I observed that MLP model containing too many hidden layers and neurons performed worse compared to its smaller counterparts, while taking more time and compute resource to train.

Through the comparison between decision tree model and MLP model (the one performs best among all 9 models), their performance to solve the binary classification problem are quite close. Hence, it is hard to identify which model performs better because the experiment is based on a specific dataset.

In the future, when selecting a model in solving similar classification problems, we should first concentrate on increasing the number of samples in the dataset as well as the number of features. Small dataset learning itself is a sophisticated sub-domain in AI and requires measurements to prevent overfitting.

If there are more samples, MLP usually has better performance for training; but if there is a relatively small amount of data containing lots of features, decision tree might perform better. Moreover, we could adopt a combination model of MLP and decision tree, which is to extract latent features using MLP with a certain number of hidden layers and parameters, and then train a decision tree classifier using said latent features to generate more accurate results. There should also be discussion around approaches to determine the optimal MLP model configuration, as shown that larger model does not necessarily translate to better performance.[6]

References:

[1] Chen, L., Gedeon, T., Hossain, M. Z., & Caldwell, S. (2017, November). Are you really angry?: detecting emotion veracity as a proposed tool for interaction. In Proceedings of the 29th Australian Conference on Computer-Human Interaction (pp. 412-416). ACM.
[2] Brownlee, J. (2020, August 15). *Crash Course On Multi-Layer Perceptron Neural Networks*. Machine Learning Mastery. https://machinelearningmastery.com/neural-networks-crash-course/
[3] S. (2019, July 21). *GELU activation - SG*. Medium. https://medium.com/@shoray.goel/gelu-gaussian-error-linear-unit-4ec59fb2e47c
[4] Gupta, P. (2018, June 20). *Decision Trees in Machine Learning - Towards Data Science*. Medium. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
[5] Brownlee, J. (2019, August 6). *How to Configure the Number of Layers and Nodes in a Neural Network*. Machine Learning Mastery. https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/
[6] *Decision Trees Compared to Regression and Neural Networks | DTREG*. (n.d.). Retrieved May 31, 2021, from https://www.dtreg.com/methodology/view/decision-trees-compared-to-regression-and-neural-networks