# Using LSTM Neural Network to predict the real or fake smile based on Observers Pupil Dataset, and apply Image Compression Technical to Pruning Neural Networks

**Chaoyun Gong**

Research School of Computer Science, Australian National University
U6329142@anu.edu.au

**Abstract.** This project will try to predict real or fake smile for human by observing pupil dilation. The neural network will be built a LSTMs neural network as the prediction model. In this model, Sigmoid activation function is used. On the other hand, the distinctiveness of image compression technical will be applied in this project's neural network to pruning the redundancy of network. This report will include four parts: Introduction of the details of dataset, networks and the distinctiveness technical. Each step of implement this project's neural network. The results and discussion for this project's neural network. And the conclusion of the whole predict model.

**Keywords:** LSTM, Sigmoid, CrossEntropy, Distinctiveness, Pruning, Neural network.

## 1 Introduction

To justify one human's smile whether is real or fake, in traditional way, we need to analyze it by our mind depends on the different situations. And it also sometimes has mistaken. However, we can apply some technical methods to distinguish the real and fake smiles. Valstar et. al. has analyzed smile videos to distinguish the smiles [1]. In another method, depend on the research, the pupil dilation and the 'size' of a lie have linear relationship [2]. Hossain, Gedeon, Sankaranarayana, Apthorp and Dawel have tried to use the process of pupil dilation when human smiles to distinguish real and fake smiles [3]. The dataset [3] is an Excel about the Asian males and females' pupil size changed when real smile and fake smile. It includes pupil dilation data of 10 Asian people's real and fake smile. The data has been normalized to 0-1. The smile video lasts 10 seconds and the sample frequency is 60 Hz. Thus, there are 600 variables of pupil size for one smile. However, because of the data loss, there are 541 variables for each smile. The dataset also includes the average pupil dilation data for males, females and all real, fake smiles. In Hossain, Gedeon, Sankaranarayana, Apthorp and Dawel paper [3], they analyze the dataset, and plot the several figures.

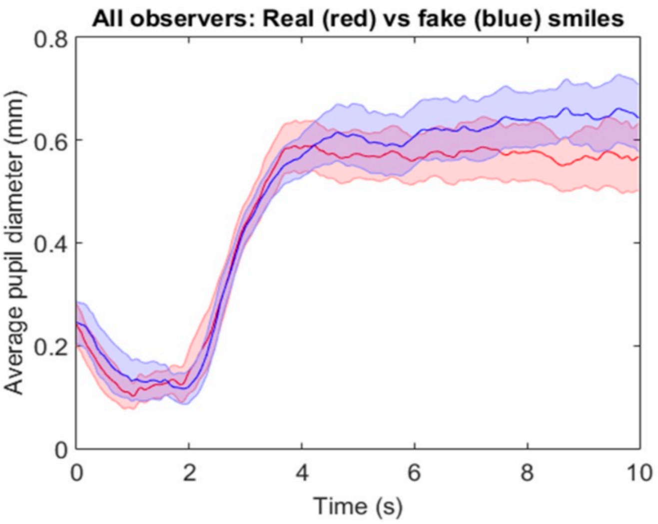



Figure 1. Average pupil diameter against the smile timeline for all observers [3]

In Figure 1, the red line is the average pupil diameter for 10 seconds real smiles timeline, the red shade is the range of all real smiles' dataset. The blue line is the average pupil diameter for 10 seconds fake smiles timeline, the blue shade is the range of all fake smiles' dataset. We can observe that there are some differences between real and fake smiles at different periods of timeline. It can be used to distinguish real and fake smiles. On the other hand, Hossain and other authors also mention that the average pupil diameter is also different for real and fake smile timeline between males and females [3]. However, this project focuses on predicting real and fake smiles regardless males and females. The details of difference between males and females will not be decrypted clearly in this report.

The reason that why we need to apply image compression technical to prune neural networks is the disadvantages of back-propagation [4]. The main disadvantage of back-propagation is that when the neural network is large, the training of network will be very slow [4]. Thus, we need apply some methods to prune networks and make the neural networks more efficient. The technical in image compression can solve this problem. The hidden layer size also depends on the radio of compression [4]. It will remove the redundant units after training [4]. Thus, we can minimize the size of network.

In image compression network, the input layer and output layer have the same size, and the hidden layer has fewer units than theirs [4]. Thus, the input means the original image, and the output means the reconstructed image. The hidden unit activations represent a compressed image. It can be called autoassociative network. There are several properties on pruning trained networks [4]. In Gedeon and Harris's paper, the property distinctiveness has been explained. And in this project, this property will also be used to prune the network for distinguish real and fake smile neural network.

# 2 Method

## 2.1 Data processing

In this project, the dataset has been updated compared with previous one. It includes two excel documents for left eye pupil dilation and right eye pupil dilation. And each excel includes 19 sheets, L1, L3, L4, L5 and H1-H5 sheets are data for real smile. A1-A10 sheets are fake smile. And in each sheet, p3 participant's data is not from Asia. however, the project focuses on Asia participants. Therefore, we need to drop it. Finally, there are 174 participants' real smile data and 196 participants' fake smile.

One problem is that there are lots of missing values in dataset, because they were not recorded properly. We cannot input Nan into neuron network. Thus, we need to fill those missing data. The data of the eye pupil dilation are sequenced and have relationship between neighbors. Then, we apply the following method to copy the upset value of the missing data and pad the empty.

```
DataFrame.fillna(method = 'ffill', axis = 0)
```

However, if there is no value before the missing pose, we fill 0.

Another problem is that the sequence length of each participant is different. The longest one is 2168 and the most participants are less than 1168. Thus, we just ignore the data which part of sequence length is over 1168.

After that, we convert the data from type DataFrame to numpy matrix. Then, we mix the real smile and fake smile dataset and split them to training set and test set. There are 300 participants (150 real, 150 fake) for training and 70 participants (24 real, 46 fake) for test.

## 2.2 Build neural network

To implement this function, predict real and fake smiles, this project tries to use LTSM neural network. Because the dataset of pupil dilation for one smile are time series, the data is a vector and they have relationships between each other. We want to a neural network which has memory and can connect one set of input data. Thus, we want to apply Recurrent Neural Networks (RNNs).

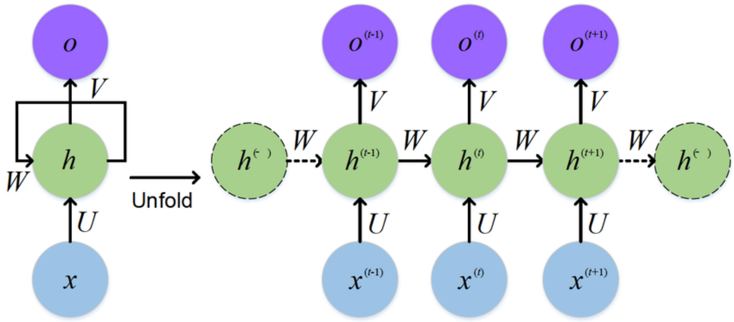



Figure 2. Recurrent Neural Networks (RNN) structure [5]

Figure 2 shows that the standard structure of Recurrent Neural Networks (RNN). The left recurrent model can be unfolded to right model, the input is sequential vector. Each previous step has a effect by weight W to next time step. Thus, the hidden state h at time t can be written as:

$$\boxed{h_t = f(Wh_{t-1} + Ux_t + bias)}$$

However, the gradient maybe explodes or vanishes when we calculate the gradient using backpropagation [6], especially when the sequence data is very long. thus, this project applies Long Short-Term Memory RNNs (LSTMs) for the dataset. The difference between them is that LSTMs use memory blocks to replace the nonlinear units in the hidden layer [6]. In memory block, there input, output, one or several memory cells and a forget gate to control the information flow in memory block [6]. The figure 3 is below to show the details of a memory block.

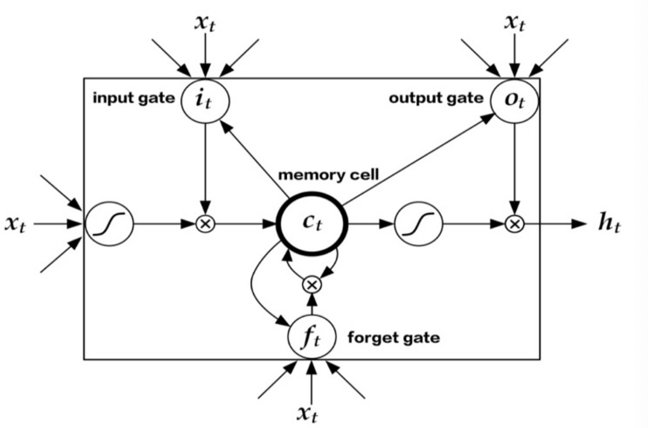
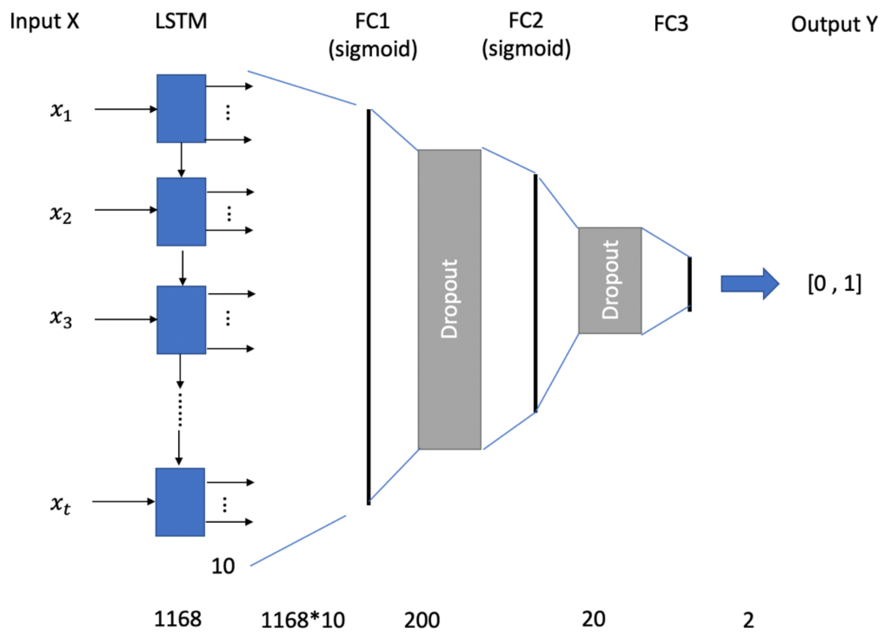



Figure 3. A memory block [6]

The advantage of LSTMs is that it can store the previous data of sequence to get better prediction and it can solve the gradient vanish problem [6].

In this project, the training dataset is a 1186*300 matrix. Firstly, we divide the dataset into four patterns to train the neuron network separately for pruning. Each pattern has a 1186*75 matrix. And we divide 75 data to 15 batches. The batch size is 5. Thus, the input of the neuron network has sequence length 1168, batch size 5 and input size 1. Then we can observe that the structure of LSTM neuron network is below:




(Figure 4. The structure of the neuron network for this project)

The structure of LSTM layer is that input size is 1, hidden size is 10, number of layers is 1. Then we use view() function of neuron network to reshape the output of LSTM to one row ([1,1168*10]). And connect the output to a full connect layer FC1 with sigmoid activation function. Activation function will be used to make the network non-linear. Then do Dropout() function to reduce the weights. After that, adding another full connect layer FC2 with sigmoid activation function. The input size is 200 and output size is 20. And also followed a Dropout() function. Finally, the last full connect layer has been added with input size 20 and output size 2. This size 2 output is the final output of the neuron network. It includes two values, one is the probability for fake smile, another is the probability for real smile.

Some hyperparameters of the neuron network are that, the epochs are 50, the learning rate is 1e-3. We also try different learning rate. However, when the rate is too large, the gradient cannot decrease all time, it will fluctuant after several times. When the rate is too small, the decrease of loss is not significant. The loss function is CrossEntropy. It is very suitable to the situation that classify the data to one class by using Softmax activation function. The optimization method is stochastic gradient descent. It is the basic and reliable method.

## 2.2 Pruning Networks

Depending on Gedeon and Harris's research [4], this project would apply image compression technical to prune the neural network. This technical is called Distinctiveness.

Distinctiveness includes four steps [4]:
Firstly, apply different patterns to train the neural network and get the output and hidden units respectively. The hidden units of patterns are outputs of unit activation function in patterns set [7]. In Gedeon and Harris's paper, the basic logistic activation function Sigmoid has been used. The equation of Sigmoid is below:

$$\boxed{y = \frac{1}{1 - e^{-x}}}$$

Secondly, arrange the data of different patterns. The variables in same position of different patterns constitute a vector. The space which includes all the vectors is called pattern space. each group outputs of the same hidden units from different patterns is a vector. Gedeon and Harris explain the vector that it represents the hidden unit's functionality in pattern space [4]. They also claim that, in this model, the units can be removed which is recognized as unnecessary with short activation vectors in pattern space.

Thirdly, recognize similarity between two vectors. The method to recognize similarity is to calculate the angle between them in pattern space. The outputs of activation function range from 0 to 1. We need to normalize them to 0.5. The method is reducing 0.5 for each output. Then, the range of normalized value represent the angle 0° to

180°. If the angle of two vectors is smaller than 15° or larger than 165°, we can remove one vector between two of them. Because 15° means they are too similar and 165° means they are complementary. The average effect is negligible [4].
In python, to calculate the angle between pair of vectors, we can apply the below equation:

$$\boxed{\cos\theta = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} = \frac{A^T B}{||A|| \times ||B||}}$$

And the code of python is below:

```
num = float(x.T @ y)
denom = np.linalg.norm(x) * np.linalg.norm(y)
cos_angle = num / denom
angle=np.arccos(cos_angle)
angle2=angle*360/2/np.pi
```

To compare each pair of vectors in pattern space, the pseudo equation is below:

$$\boxed{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{{x_i}^T x_j}{||x_i|| \times ||x_j||}}$$

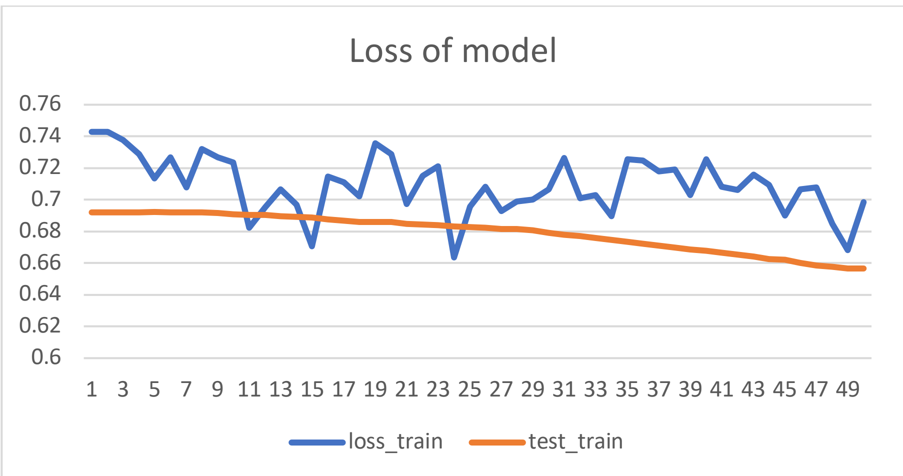
However, all value will not be summed, there will be stored in a list. In python code, we can use two for loop to implement it. The pseudo code is below:

```
for i in range((len(param[0])-1)):
    for j in range((len(param[0])-1-i)):
```

Finally, the new weights of previous neural network need to be adjusted. However, the neural network does not need to re-train [6].

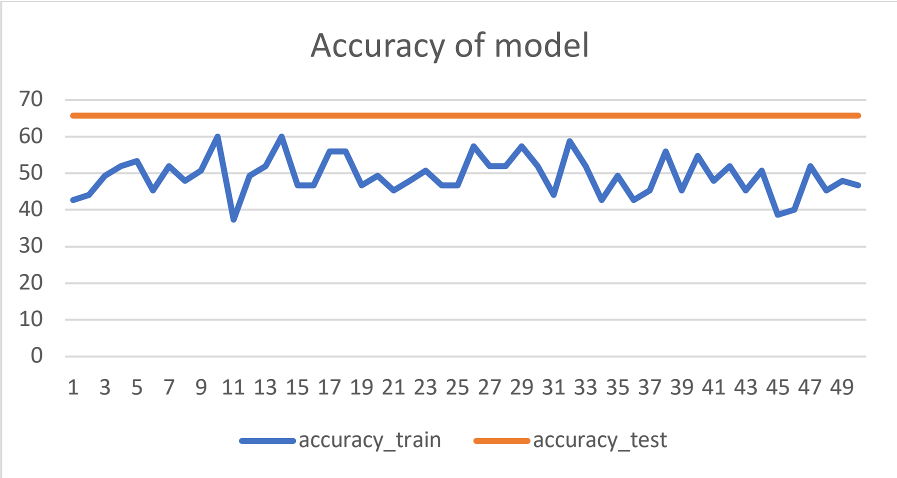
# 3 Results and Discussion

The epoch of the neural network is set to 50. Each epoch includes 15 batches. They all include real and fake smiles dataset respectively. The training and test loss of neural network for the 10 epochs is below:




(Figure 5. The loss of neural network)

Figure 5 show that the loss of this project's neural network for epoch 1 to 50. Blue line is the loss of training dataset, orange line is the loss of test dataset. The problem is that the loss of training dataset decreases at the first several epochs but is fluctuant after that. The reasons are possibility that the learning rate is too large, the structure of neural network is not suitable, or the dataset has problem. For orange line, the loss of test dataset, it decreases slowly and keeps stable at the end of 50 epoch. The tendency of test dataset's loss is correct. However, the range of the decline is too small. It means that the improvement of the test dataset's performance is very small.




(Figure 6. The accuracy of neural network)

Thus, we can observe that the accuracy of neural network for training and test dataset in Figure 6. The blue line is the accuracy of training dataset. It is similar to the loss of training dataset and fluctuates at all time. The accuracy of test dataset always equals to 34.29%. The reason is that the decline of the loss of model is too small. It means that the model does not have good performance for this dataset.

For distinctiveness, in Gedeon and Harris's paper, the example of distinctiveness is below [4]:

| Pattern | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| p.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.000 |
| p.001 | 0.000 | 1.000 | 1.000 | 0.000 | 0.000 | 0.000 |
| p.002 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 1.000 |
| p.003 | 0.000 | 0.000 | 0.000 | 1.000 | 1.000 | 0.000 |
| p.004 | 0.000 | 0.000 | 1.000 | 1.000 | 0.000 | 0.000 |
| p.005 | 1.000 | 0.439 | 1.000 | 1.000 | 0.999 | 0.706 |
| p.006 | 0.000 | 1.000 | 0.000 | 1.000 | 1.000 | 0.000 |
| p.007 | 1.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 |
| p.008 | 1.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 |
| p.009 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| p.010 | 0.000 | 0.000 | 1.000 | 0.000 | 0.167 | 0.000 |
| p.011 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 |
| p.012 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| p.013 | 1.000 | 0.000 | 0.000 | 0.989 | 1.000 | 1.000 |
| p.014 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| p.015 | 0.000 | 0.000 | 1.000 | 1.000 | 1.000 | 0.000 |

Table 1. Six hidden unit activations by pattern.

| Pair of units | Vector angle |
|---|---|
| 1 2 | 71.7 |
| 1 3 | 90.0 |
| 1 4 | 68.0 |
| 1 5 | 68.9 |
| 1 6 | 61.6 |
| 2 3 | 71.7 |
| 2 4 | 94.1 |
| 2 5 | 64.5 |
| 2 6 | 70.9 |
| 3 4 | 97.1 |
| 3 5 | 81.5 |
| 3 6 | 92.2 |
| 4 5 | 60.9 |
| 4 6 | 84.9 |
| 5 6 | 70.6 |

Table 2. Vector angles for pairs of the hidden units.

The table 1 shows 16 dimensions pattern space with 6 vectors. And the table 2 shows the angle between each pair of vectors. There are all between 60° and 100° and no pair similar or complementary.

In this project work, we sample one full connect layer. The pattern space has 4 dimensions, and the hidden units includes 20*200 vectors. Because of the large size of data, the vectors will not be shown in report. The result is that in the 20*200 vectors, after many times calculation of pair angle, only 7 vectors are left, the other units have been removed because of similarity or complementary.

After normalizing to 0.5. Each unit has 4 dimensions. We can observe that the value of vector's each dimension is very small. It may have insensitive effect in this project's neural network.

Applying the test dataset, the performance of the new model (after pruning) is that the loss of new network the loss is 0.689, the accuracy is 65.714%. The accuracy is a little bit higher than the original model (0.656) and the accuracy is the same. So, removing the redundancy units of one full connect layer in this project network does not have large effect of the performance of model. Thus, we can apply this technical for our model. It can reduce the complexity of neural network dramatically and will not affect the accuracy of neural network.

## 4 Conclusion and Future Work

This project tries to predict the real and fake smile depend on people's pupil dilation by using LSTMs neutral network. The final model is not suitable for the new dataset. The LSTMs neutral network should be modified in further.
On the other hand, we just use the distinctiveness technical in one full connect layer in this project to prune the network. And it produced the expected results. When the technical of image compression in Gedeon and Harris's paper can be applied more widely in this neural network, we can build a more complicated and accurate neural network and make it more efficient by pruning the network.

## 5 References


1. Valstar, M. F., Gunes H., Pantic, M. (2007). How to Distinguish Posed from Spontaneous Smiles using Geometric Features, Proceedings of ACM International Conference on Multimodal Interfaces (Nagoya, November 2007), 38-45.
2. Wang, J. T.-y., Spezio, M., Camerer, C. F. (2009). Pinocchio's Pupil: Using Eye tracking and Pupil Dilation To Understand Truth-telling and Deception in Sender-Receiver Games, American Economic Review, 100(3), 984-1007.
3. Hossain, M.Z., Gedeon, T.D., Sankaranarayana, R., Apthorp, D., Dawel, A. (2016). Pupillary Responses of Asian Observers in Discriminating Real from Fake Smiles: a Preliminary Study. Proceedings of Measuring Behavior 2016, (Dublin, Ireland, 25-27 May 2016).
4. Gedeon, T.D., Harris, D. (1992). Progressive Image Compression. [Proceedings 1992] IJCNN International Joint Conference on Neural Networks. DOI: 10.1109/IJCNN.1992.227311
5. Feng, W., Guan, N., Li, Y., Zhang, X., Luo, Z. (2017). Audio visual speech recognition with multimodal recurrent neural networks. 681-688. 10.1109/IJCNN.2017.7965918.
6. Plested, J. (2019) COMP8420 Neural Networks, Deep Learning and Bio-inspired Computing. Sequence Learning part1.
7. Gedeon, TD, Harris, D, (1991). "Network Reduction Techniques," Proc. Int. Conf. on Neural Networks Methodologies and Applications, AMSE, San Diego, vol. 2, pp. 25-34, 1991.