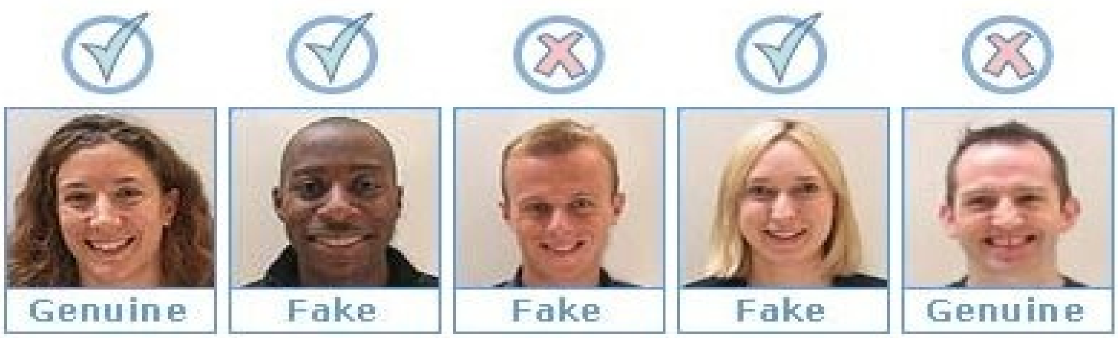
Deep Learning(CSL4020)

DLOPS PROJECT REPORT

Real vs Fake Smile Detection



Introduction

As the paper title suggests, the task is distinguishing between real and fake smiles. In a naive way, our brain analyzes the smile of the person whether it is real or fake depending upon the context and circumstances.But at times, it fails to distinguish between a real and a fake smile. Nowadays, pattern recognition and facial expression recognition are quite trendy topics and open to research. Since there can be many varieties in the facial structure of humans, hence smile detection can be considered as a tough task.

The final aim of our project is to design a deep learning model which can distinguish fake smiles from real smiles when an image of a smiling person is passed as input to this model. This problem can be solved by analyzing the smiling images of the person. The researchers have stated the amount of magnitude of lie and the pupil dilation have a linear relationship between them. So, we are basically studying the impact of smiling on the pupil dilation and utilizing it to predict the output.

Method

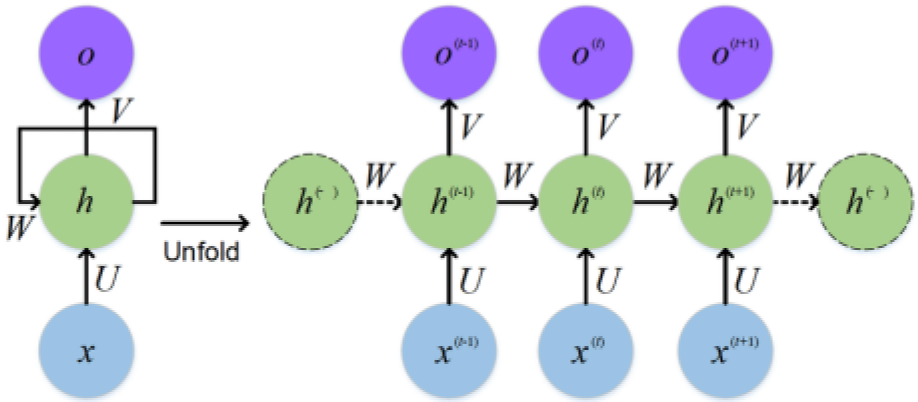
So, let us first discuss the data processing which is to be done on the dataset provided with us.

Data processing

The dataset has two further subdivisions for the left eye dilation and the other data is available for the right eye dilation. Also, these data points are further divided into real smiles and fake smiles which was the aim of our project. So, this dataset will help in training the model easily. Also, there is an issue that some values in the dataset are missing. This is a common problem which might occur due to mistakes in collection of data. As we have discussed that the value of data points for eye dilation are in a sequence and they have correlation between them, we can utilize this property to our advantage and find out the missing value and fill it.

Neural Network

As we have seen in the above dataset, the values for the pupil dilation are very much in sequence and correlated with its neighbor. So, to train this kind of data, we require a neural network which has a certain amount of memory and can relate or connect to one set of input data. The description fits for a certain kind of neural network, namely Recurrent Neural Networks(RNN) and we utilize it to train this kind of time series data for pupil dilation.

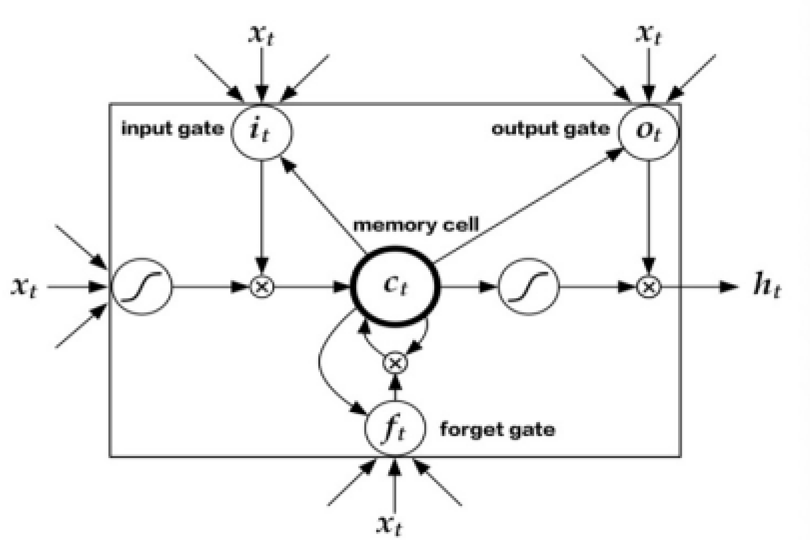


The above mentioned structure is a standard or textbook structure of Recurrent Neural Networks.

Also, in the diagram above we can see two separate structures distanced by an arrow mentioning unfold and the right side is just an extension of the diagram which is shown on the left side and the input as shown is a time series or sequential input which fits suitably for our case. So, all the variable values or weights which are considered are shown on the diagram. We can utilize the above structure to calculate the hidden state at time t to calculate:

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

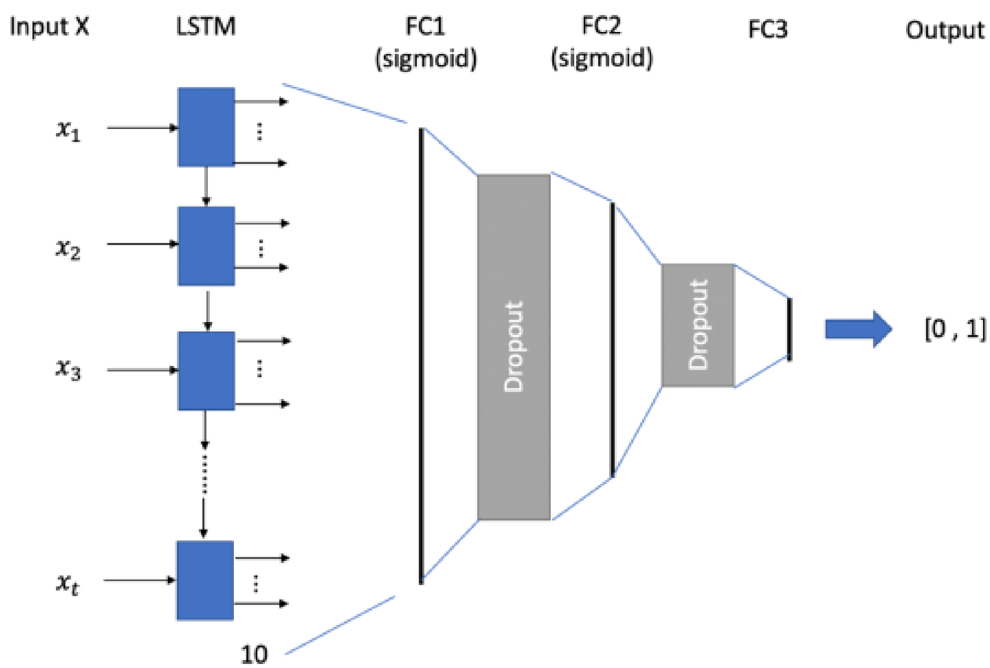
There may be an issue arising with the value of the gradients. When we apply back propagation on it, we get a case which involves the problem of explosion or vanishing gradients as the length of neural networks is very large. We have studied in class that if the length of sequence data is very long, this would lead to weight updation being very less impactful and hence the problem of vanishing gradients would come into picture. So, we are employing the usage of Long Short-Term Memory RNNs for our dataset of sequential value of eye dilation. We know that these issues are resolved in LSTM because of the usage of memory blocks for the LSTM. The memory block of LSTM consists of input, output of one or memory cells and it also consists of a forget gate which has the sole purpose of determining which particular data to be left out, i.e., it controls the information flow in this structure. The major advantage of LSTM and the bigger reason behind selecting it for our model is that it is designed to work efficiently with sequential data, that is it can utilize the previous chunks of data to train itself and also, solve the problem of vanishing gradient problem.



The memory block of LSTM structure is shown above and is described in the above paragraph and also the forget gate is an important component of it.

Workflow

The workflow or the architecture which will be implemented by us in our code is given below and then, let us understand the structure in detail about it and finally talk about the results which we have obtained on running the dataset on this model.



Workflow

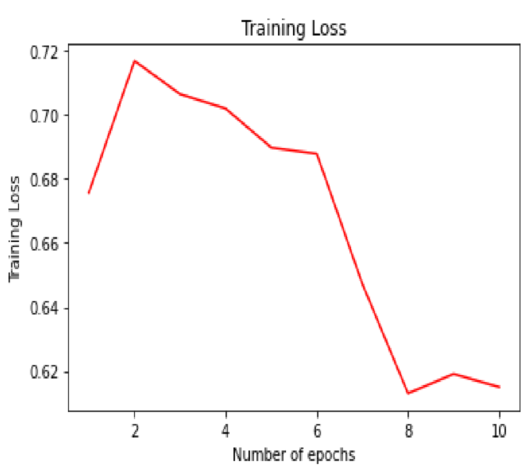
So, as we can see in the above diagram, firstly the input is passed to the LSTM which takes 1 parameter as input and the output size is 10 and number of layers for this arrangement is 1. So, we have t number of inputs to the LSTM then for every block, there will be 10 output and hence the total output will 10*t. So, we will be using the view() function to reshape the output such that all these outputs from each LSTM will be combined in fashion that the size now becomes equal to [1,t*10]. This is then passed as input to the fully connected layer FC1 and sigmoid activation function is used to make the architecture non-linear. As can be seen in the diagram, dropout is added which will remove the weights. Then, this will be passed to the fully connected Layer 2(FC2) with the sigmoid activation layers. It takes 200 as the input size and the size or the number of neurons after the FC2 layer is 20. Then, again a dropout function is added to remove some weights which will reduce a certain number of parameters. Since the output size was 20, the input size to the 3rd FC layer will be 20 and the output size will be 2 which consists of a real or fake smile which must be our final output. So, this final 2 size layer is the final output for our model and as mentioned earlier, it consists of probability for a fake smile and the other one consists of the probability of real smile.

Hyperparameters

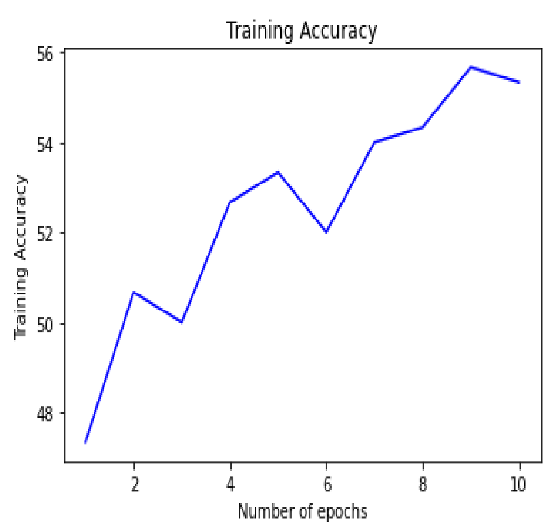
Some of the hyperparameters for our training of neural network models are the number of epochs as 10 and the learning rate as 0.001.So, we can also check for other learning rates and then compare the results which we are getting. The loss function for our model is CrossEntropy and Softmax activation function is being used for our case. The optimization method which is followed for our model is stochastic gradient descent. We have to select the value of these hyperparameters in such a way that we can optimize it.

Results

Training Loss



Training Accuracy



Due to lack of computational power, only limited epochs could be performed. As you can see from the graphs, the trend of training loss and accuracy vs the number of epochs is decreasing and increasing respectively. The test accuracy came out to be 62.8571%.

Conclusion

As the project requires very unique facial features like pupil dilation and smile expression, any model would be difficult to train on the basis of this dataset. The accuracy we got is indeed a decent number and very difficult to achieve. Further modifications can be done to improve the accuracy but it would require high computational power.

References

https://drive.google.com/drive/folders/1YZj1F3MhD7kdyc2LBm4YZYPZK1giAIk2

https://www.researchgate.net/publication/322877298_Discriminating_real_from_fake_smile_using_convolution_neural_network

https://ieeexplore.ieee.org/document/9023790

By - Sarthak Vasan (B19BB040)

Shivam Zade (B19BB043)

Harsh Kumar (B19CSE036)