Deep Neural Network approach to Static Facial Expression Recognization

Hemang Jethava, Jainil Vachhani, Nand Parikh, Palash Hariyani, Parth Shah

School of Engineering and Applied Science
Ahmedabad University

***Abstract*—Facial expressions of any person can be determined on a computational device viz. computers. A deep learning approach to this problem can be developed using concepts of sequential models and convoluted neural network(CNN) or convnets. We have proposed a deep learning sequential model comprised of CNN, Maximum pooling, Rectified linear Units and fully connected layers.Applications of discussed topic extends from simply facilitating retailers to analyse customer satisfaction to more complex encounters like guessing patients mental state during treatment or estimating audiences reaction to a play or movie.**

Keywords: Connvolution Neural Networks(CNN), Pooling, Rectified Linear Unit(ReLu).

I. Model for Deep Neural Network

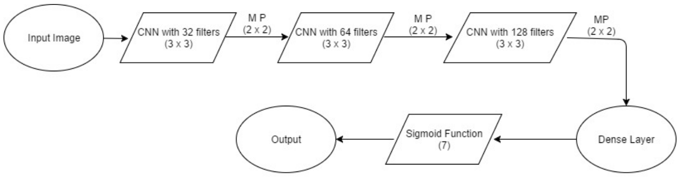
The mentioned problem of facial expression recognization has been sufficiently explored by [1], where [1] has proposed a deep sequential model using convnets. The model proposed here is a replica of the model presented in [1], with minor modifications. Fig-1 shows the model developed. Each CNN has different number of hidden layers, feature matrix in each of the convnet is kept of size 3x3.Maxpooling on the image is done using a matrix of size 2x2.Activation function for each is Rectified linear unit(ReLu). Explaination of each of the above topic can be found in [2]. [3] has provided a comprehensive dataset consisting of about $40,000$ images which has been used here.

II. Heuristics for DNN

There are many hyperparameters like *batch size, epoch, learning rate, validation set, number of hidden layers, neurons in each layer etc.* that affect the accuracy of DNN. Heuristics for these hyperparameters provides us certain ranges of the above mentioned parameters under which a model performs better in genral. We can always use a trial and error method and expect a better accuracy for our network. A comparision of various parameters and it's effect on accuracy and computational time is further discussed.

III. Comparision of Hyperparameters

All the hyper parameters mentioned in heuristics were analyzed. Keras library used provides an inbuilt regression optimizer *adam*, which assures a good learning rate. [1] has provided a good analysis for hidden layers and number of neurons in each of the layers. [4] has also suggested to use a *single validation fold to cross validation*. A 20% single validation fold has been used. Table-1 and Table-2 has been generated for various values of batch size and epoch, for both the cases 20% *validation_split, model (fig-1) and 28,709 training pairs* were constant. For variation in batch-size epoch was 55 and for variation in epoch batch-size was 128.Tables indicate that at higher epochs and higher batch-size accuracy was higher, but computational time also increased with it. After a certain number of epochs and batch-size accuracy is not changed much, this can be found using trial and error.


Input Image
CNN with 32 filters (3 x 3)
M P (2 x 2)
CNN with 64 filters (3 x 3)
M P (2 x 2)
CNN with 128 filters (3 x 3)
MP (2 x 2)
Dense Layer
Sigmoid Function (7)
Output

M P = Max Pooling

Fig-1:Model for DNN

Batch-Size	Accuracy	Time taken(min)
16	50.58	12
32	69.28	33
64	74.43	76
128	87.88	94

Table1: Accuracy and Time for Batch Size

Epoch	Accuracy	Time taken(min)
1	33.114	4
10	63.17	35
55	85.46	94
100	94.37	238

Table2: Accuracy and Time for Epoch

References

[1] https://github.com/JostineHo/mememoji

[2] http://brohrer.github.io/how _convolutional _neural _networks _work.html

[3] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge

[4] http://cs231n.github.io/neural-networks-3