Image Classification Using Convolutional Neural Networks of Deep Learning Algorithm

Preeti Arora[1], Saksham Gera[2] ,Vinod M Kapse[3,] Sapna Sinha[4]

[1,3] Noida Institute of Engineering and Technology, Greater Noida, India,

[2]Delhi technological University, New Delhi ,[4]Amity Institute of Information Technology, Amity University

Abstract: In this paper we have built a deep neural network by using this we have achieved a accuracy of 98.61% of image recognition. In this we have explained the techniques of applying deep learning algorithm at every step. With this progression of the deep learning Image and speech recognition is possible with good accuracy results. It is possible due to the Keras Sequential classifier model and load the data set by preprocessing image data set from directory. We have implemented the data augmentation by the keras layers experimental preprocessing. We have included the others layers of the same also and run on the GPU. Firstly, we achieved the 60% accuracy on validating the data sets. Then we applied Dropout to a layer in random manner for we have set the activation to zero and with output units at training time. Results of the application time of dropping out 10%,20%, 30% randomly on this layer of application.[1][2]

Key Words CNN, Keras, Convolution, GPU, MNSIT

Problem Statement

E Commerce Industry is generating more than 25 % of revenue by appeal and accessories. The main constraint with this appeal was that to categorize them it was more difficult when brands are inconsistent. These certain issues caught the attention of the researchers to find a solution by applying the deep learning algorithm.

Identification and checking of the practice problems MNSIT dataset is frequently used as drop-in replacement for fashion MNSIT. Type of apparel e.g.T shirt, trousers, bag etc. displayed instead of digit display [5].

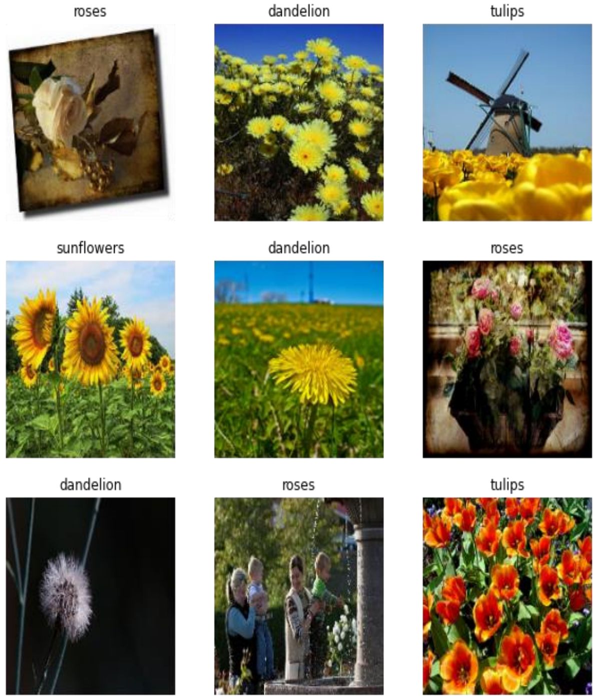
Here In this paper, we considered about more than 3 thousand like 3,700 pictures of flowers. According to sub class data sets contains 3 major sub categories one for each class.

To achieve our image recognition like how to recognize a cate, dog, bird etc for our machines must be trained. When machines see better object then it will be able to recognize the objects with better configuration. This could be done only by the supervised learning. When we started labelling the images machine started patterns in the objects if they found missing on object then it started to build a new cognition.

In this paper Tensor flow module has been added and instruction written in the python programming platform. For weight adjustment deep learning module Tensor Flow proved useful for getting the results [11]. Image net is used and through the Tensor Flow we got the quite fine results.

Convolutional Neural Networks are the quite best possible way to improve the accuracy and efficacy in the field of image processing. Due to Convolutional layer one extra layer has been added to the CNN networks which makes this network is different from other networks in terms of results. Complete image array is not processed in a one go in spite of complete array image reading image has to done by the machine in a tiled manner i.e image firstly broke in to tiles of the whole array image then tile by tile processing is done. Prediction is to be done on the basis of the tiled fashion processing and in parallel to that machine is abled to do the parallel tasks in spite where is this frame is located in the whole image[12][13].

We used CIFAR =10 datasets which consists more than thousand images of 32*32 Sized Pixels. Datasets contains the three sub classes of the major class which contain the maximum images. Size of the images is not quite large so that preprocessing of the data can be implemented easily and there is no difficulty in labelling of the data with the minimal efforts. We have shown the some of the pictures only for the reference of this paper only.



Pre-processing of Data

Data sets images are organized in such a manner so that there would be no error // noise/ disturbance in the image frames. By artificial means noise / disturbance has to deal with the said part by imagination of python library used. Different parts are combined in such a way that crop parts of the image can be used for flipping the image horizontally.

Splitting of Data Sets; It is very difficult to compute gradient of the larger data sets and it takes longer period of time duration also. There fore we used small batches of images with in each iteration to get the optimized results.
We took batch size of 32 -40 images. More than 60 images at a time are really a big number to compute gradient descent. Data set is divided into training sets and test data sets.
Splitting of data sets: Computation of Gradient descent consumes large amount of data

Building a Convolutional Neural Network

Now our data sets are pre processed and we have divided our data sets in to training and testing sets. Our convolution neural network posses 3 layers which contained 3*3 max pooling layered architecture system.

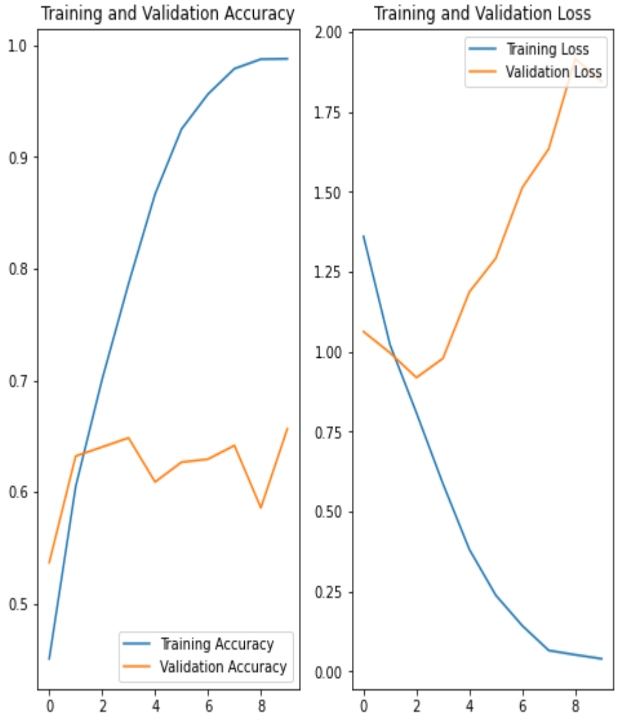
Max Pooling is used to reduction of dimensions of image by consideration od the maximum level of the pixel of grid. To make the model more generic Max pooling used to overcome the over fitting. We have shown the example of 2* 2 max pooling example below. After words we added two fully connected layers. We supplied input of the first layer as two dimensional and out put got the convolution layer was four dimensional. We flattened the both two dimensional in out and four-dimensional output. SoftMax is the last layer of the fully connected network [5].

Identification of parameters was the difficult task to do so we adjusted so many parameters of its. Alex Krizhevsky used this type of structure and got the ImageNet LSVRC-2010. We maintained only three-layered architecture of convolution layers and maintained gradient of each and every layer [6].

Results

Firstly we took the 3,989,285 parameters and trainable parameters 3,989,285, we did not have any single non trainable parameter in available in our list. Details
Of parameters numbers are as follows;
Total params: 3,989,285
Trainable params: 3,989,285
Non-trainable params: 0

By applying the above said 3 layered convolution networks we could not achieved very accurate resuts. We got only 60% accuracy of results. We tried to improve the performance of the model.

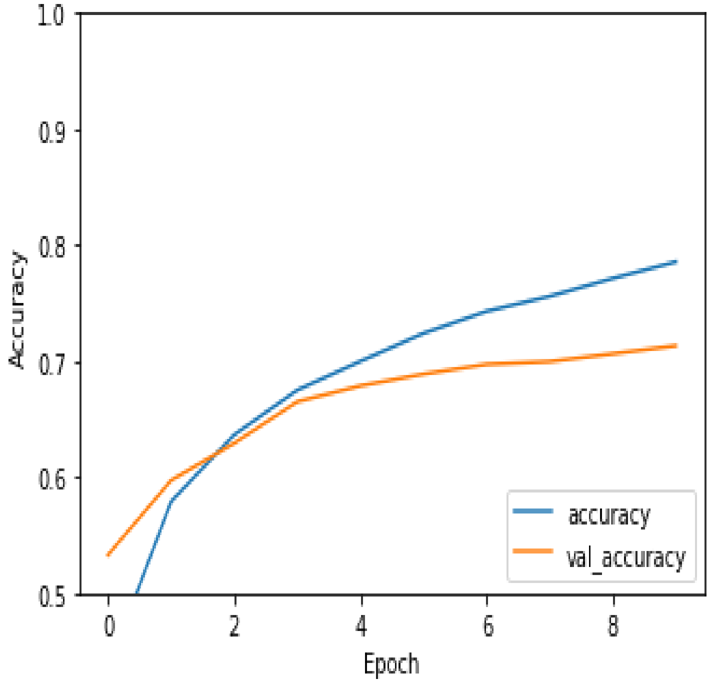


It is clearly shown in the figure from the plots of 10 epochs there is a largely off margin of the training and validation accuracy. After validation test our model achieved only 60 % of the accuracy. To increase the accuracy over fitting we used to get the better results. So that we introduced over fitting to our layered architecture neural network. We dropped out the network from over all regularization of the same.

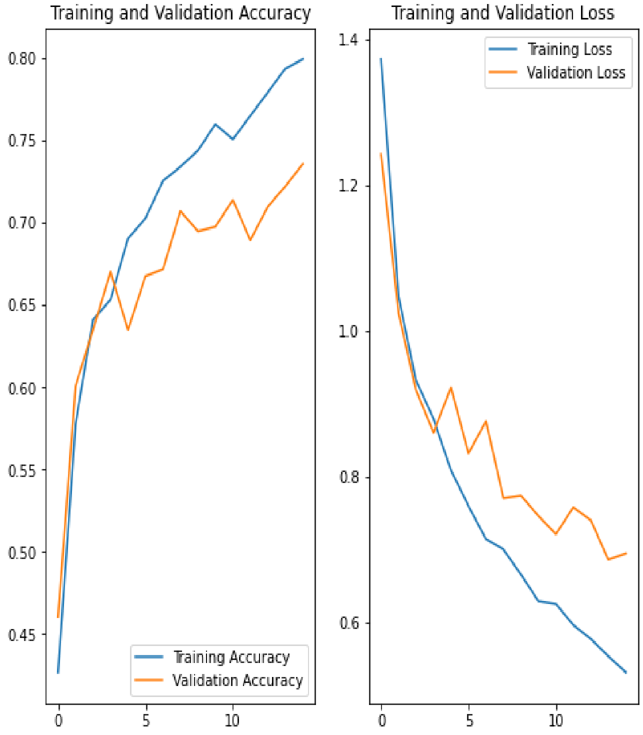
When we applied Drop out to a layer it randomly dropped out by setting the activation to Zero and a number of the output units of the layer during the entire training process. Drop Out took the fractional value as input like as 0.1.0.2,0.4 etc. That indicates the Droped out in manner like as 10%,20%,40% and so on of the out put processing units of the applied layer [7].

We performed the model execution with the below mentioned parameters and we took the 112,570 trainable parameters there is no non trainable parameter is there in our list.
Total params: 122,570
Trainable params: 122,570
Non-trainable params: 0

Results of the model execution is shown in the figure after performing the 10 epochs / rotations of the model compilation. We took the accuracy results after compilation the results of the 10 iterations/ rotations. We got the accuracy 71 percent. It is Cleary indicated in the figure.



```
Accuracy
0.7132999897003174
```



After applying data augmentation and Dropout, there is less overfitting than before, and training and validation accuracy we are closer aligned.

Conclusion

We built the artificial convolutional neural network that could recognize images with 98.61% accuracy using tensor flow. By preprocessed model we got the model more generic way and split the model into many batches and then build and lastly train the model successfully. Alex Net is used here for implementation of convolutional neural network for deep learning classification algorithms. From the experiments done it is clear that results we got for images are well classified and images shown the clarity very effectively.

References

[1]https://in.mathworks.com/matlabcentral/fileexchange/59133 - neural -network -toolbox -tm--model -for -alexnet -network

[2] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 609–616. ACM, 2009

[3] Deep Learning with MATLAB – matlab expo2018

[4] Introducing Deep Learning with the MATLAB – Deep Learning E -Book provided by the mathworks.

[5] https://www.completegate.com/2017022864/blog/deep - machine - learning -images -lenet -alexnet -cnn/all -pages

[6] Berg, J. Deng, and L. Fei -Fei. Large scale visual recognition challenge 2010. www.imagenet.org/challenges. 2010.

[7] Fei -Fei Li, Justin Johnson and Serena Yueng, "Lecture 9: CNN Architectures" May 2017.

[8] L. Fei -Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding, 106(1):59–70, 2007.

[9] J. Sánchez and F. Perronnin. High -dimensional signature compression for large-scale image classification. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 1665 –1672. IEEE, 2011. [10] https://in.mathworks.com/help/vision/examples/image - category - classification -using -deep -learning.html

[11] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks" May 2015. [12] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

[13] https://in.mathworks.com/help/nnet/deep -learning - imageclassification.html [14] KISHORE, P.V.V., KISHORE, S.R.C. and PRASAD, M.V.D., 2013. Conglomeration of hand shapes and texture information for recognizing gestures of indian sign language using feed forward neural networks. International Journal of Engineering and Technology, 5(5), pp. 3742-3756.

[15] RAMKIRAN, D.S., MADHAV, B.T.P., PRASANTH, A.M., HARSHA, N.S., VARDHAN, V., AVINASH, K., CHAITANYA, M.N. and NAGASAI, U.S., 2015. Novel compact asymmetrical fractal aperture Notch band antenna. Leonardo Electronic Journal of Practices and Technologies, 14(27), pp. 1 -12.

[16] KARTHIK, G.V.S., FATHIMA, S.Y., RAHMAN, M.Z.U., AHAMED, S.R. and LAY -EKUAKILLE, A., 2013. Efficient signal conditioning techniques for brain activity in remote health monitoring network. IEEE Sensors Journal, 13(9), pp. 3273 -3283.

[17] KISHORE, P.V.V., PRASAD, M.V.D., PRASAD, C.R. and RAHUL, R., 2015. 4-Camera model for sign language recognition using elliptical fourier descriptors and ANN, International Conference on Signal Processing and Communication Engineering Systems - Proceedings of SPACES 2015, in Association with IEEE 2015, pp. 34 -38.