Optimizing Convolutional Neural Networks for Accurate Image Prediction

Daniel Thompson*[1] & Emily Johnson[2]
***[1] Department of Computer Science, University of Oxford, Oxford, UK**
[2] Department of Electrical Engineering, University of Cambridge, Cambridge, UK

ABSTRACT

Image classification is a complex process that may be affected by many factors. There are supervised and unsupervised classification techniques. The emphasis is placed on the deep neural network classification approach and how this technique is used for improving classification accuracy. In this dissertation work we are working on the image prediction using deep learning method. A lot of methods available for the image detection and prediction. Some of the methods are static and some are adaptive methods. We're use convolutional neural networks (CNNs) to perform our task of image detection using deep learning. We're going to try to create a deep learning CNN model for an old Kaggle completion called Dogs vs Cats. There are more than 25000 images of cats and dogs are available for training purpose and 12,500 in the test set that we have to try to label for this dissertation work. Out of which we are using a data set of 2000 samples for training purpose and choose 200 images (100 of each) for testing purpose and finally checked that how our network is performing.

Keywords: *Convolution neural networks, dataset, training, testing, validation error*

I. INTRODUCTION

Image classification is a complex process that may be affected by many factors. There are supervised and unsupervised classification techniques. The emphasis is placed on the deep neural network classification [2] approach and how this technique is used for improving classification accuracy.

Cabral et al (2011) [1] described classification of remote sensing data is used to assign corresponding labels with respect to homogeneous characteristics of groups. The main aim of classification is to discriminate multiple objects from each other within the image. Classification will be executed on the base of spectral or spectrally defined features, such as density, texture etc., in the feature space. It can be said that classification divides the feature space into several classes based on a decision rule. Figure 1 shows the concept of classification of data. In many cases, classification will be undertaken using a computer, with the use of mathematical classification techniques. The learning algorithms are broadly classified into supervised and unsupervised learning techniques. The distinction is drawn from how the learner classifies data. In supervised learning, the classes are predetermined. These classes can be conceived of as a finite set, previously arrived at by a human. In practice, a certain classes of data will be labeled with these classifications. M. M. Cisse (2013) [3] reviewed the classes are then evaluated based on their predictive capacity in relation to measures of variance in the data itself. Some of the examples of supervised classification techniques are Back Propagation Network (BPN)[4], Learning Vector Quantization (LVQ)[5], Self Organizing Map (SOM)[6,7], Support Vector Machine (SVM)[8,9], etc., Here we will use the convolution neural network technique for classification.

Machine learning is what outlines it out, and the outline is something like this that a computer program is said to learn from certain experience E with respect to a certain class of task T and a performance measure P. So, if we see there are three attributes to this activity of what is called as learning or what he calls has learned. So, there is one factor which is called as an experience E there is a particular task which it has to perform T and there is a performance measure P.

Now, in order to solve this task we were gaining certain experience E, and that is by looking into multiple number of images. So, the more the images I looked through the more is the experience the more the number of epochs over, which I translate the more is the experience over there. Now from this basis of T and E we were using a certain measure called a P and this performance measure P which we were using over here, there is something which is our error function. So, that was the cost function which we were using down over there, now as we see that as our experience was increasing which is our number of epochs over which we were translating and the number of images, which we were looking down over there our error was coming down and; that means, that we somehow able to measure our performance and see that the performance is increasing. So, as the performance increases it becomes more and more accurate and accordingly our error keeps on going down ok. So, that is what is saying that if it is it said to learn if this performance on a task T which is of our classification as measured by this performance P improves with experience and now this definition if we really try to introspect onto this one, this definition is in no way very different from how human learning is all centered around. In fact, human learning is also quite similar there also, as human beings when we say that we are learning about something then the whole task of learning is when we are able to really getting more and as we get more and more experience and that is what practice makes we mean man perfect so, with that one. So, we get more experience and then we r measure becomes higher and higher with respect to a certain class of task itself. So, that is what goes down by the very classical standard definition.

II. OBJECTIVE OF OUR WORK

In our real life applications sometimes we have some images which are blurred or noisy and they are not recognizable. By choosing the appropriate deep learning tool we can predict the lost information of the image and classify it that the image belongs to which category? CNN use the following methodology to solve the problem of deep learning.

- Dense connections
- Parameter allotment equivariant, and
- Representations.

Moreover, convolution provides a means for working with inputs of variable size.
We have seen the working procedure of convolution neural network. Now we are applying the CNN for the detection of Cat vs Dog. Here we have a data set of 2000 cats and dogs (1000 images of cats and 1000 images of dogs). This data set has downloaded from “Kaggle”. We have collect validation set of 200 samples a test samples are also of 200 size. In test dataset 100 images of cats and 100 images of dogs are there. Test criteria for image detection are average loss and accuracy. Here we are also calculating the validation loss and validation accuracy.

III. INTRODUCTION TO OUR FIRST TASK: 'DOGS VS CATS'

We're use convolutional neural networks (CNNs) to perform our task of image detection using deep learning. We're going to try to create a deep learning CNN model for an old Kaggle completion called Dogs vs Cats. There are more than 25000 images of cats and dogs are available for training purpose and 12,500 in the test set that we have to try to label for this dissertation work. Out of which we are using a data set of 2000 samples for training purpose and choose 200 images (100 of each) for testing purpose and finally checked that how our network is performing.

IV. ARCHITECTURE OF CNN FOR IMAGE DETECTION

Step 1 – Convolution
Design the 2D convolution neural networks having the input shape of size 64 x64 x 3 and the activation function is Relu.
Step 2 - Pooling
Choose the pooling size of 2x2
Step 3 - Adding a second convolution layer
Add a second convolution neural network of size 32x3x3. In the second layer of network the activation function is Relu and the pooling sixe is 2x2 remain same.
Stage-4 Flattening
Flattening is the process to flatten the CNN architecture.
Stage -5 Full connection
The output dimension of the fully connected network is the 128 and activation function is relu.
Finally we take the single output and output activation function is 'sigmoid'.
Stage - 6 Compiling the CNN
For the compilation of the network we have ‘adam’ is the optimization algorithm and the loss function is 'binary cross entropy'. We have choose the target size is (64, 64), batch size is 32 and Class mode is 'binary'. Target size and test size parameters are keeping same.

V. PYTHON LIBRARIES AND SETUP FOR DEEP LEARNING PROGRAM

Here we are using the following python libraries.

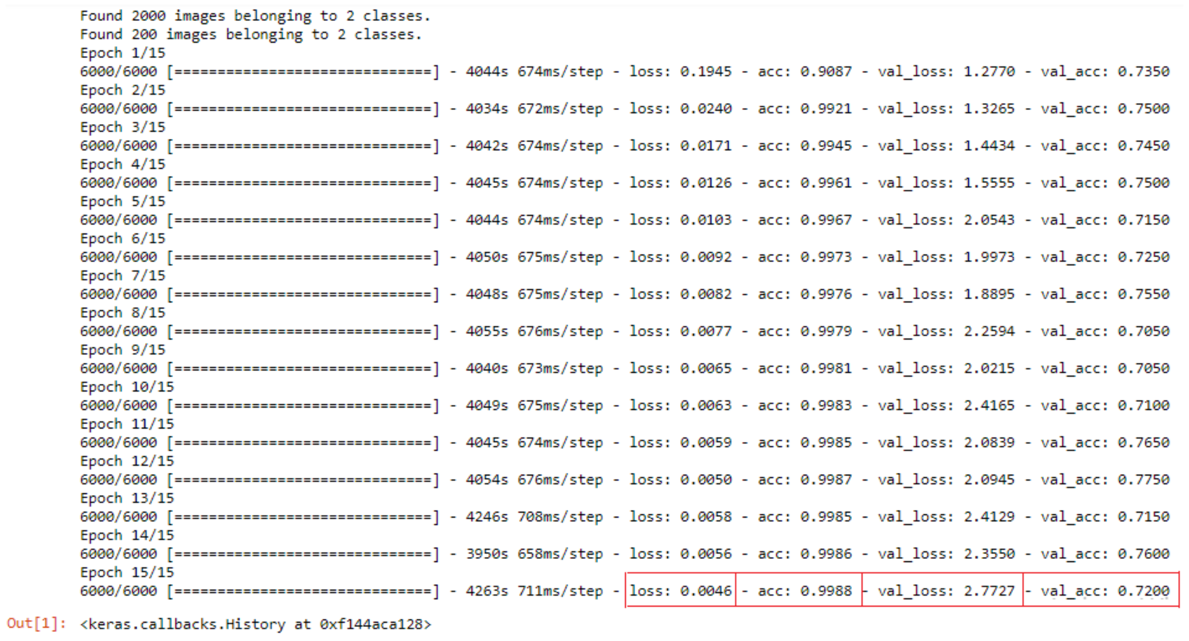
- KERAS
- TENSORFLOW
- THEANO

Libraries of KERAS used in program

- Sequential
- Convolution2D
- MaxPooling2D
- Flatten
- Dense
- PlotLossesKeras

VI. OUTPUT RESULT

To run the program we have choose the 6000 samples per epoch and there are total 15 epoch in our program. For the above parameter the executed results are as follows.



```
Found 2000 images belonging to 2 classes.
Found 200 images belonging to 2 classes.
Epoch 1/15
6000/6000 [==============================] - 4044s 674ms/step - loss: 0.1945 - acc: 0.9087 - val_loss: 1.2770 - val_acc: 0.7350
Epoch 2/15
6000/6000 [==============================] - 4034s 672ms/step - loss: 0.0240 - acc: 0.9921 - val_loss: 1.3265 - val_acc: 0.7500
Epoch 3/15
6000/6000 [==============================] - 4042s 674ms/step - loss: 0.0171 - acc: 0.9945 - val_loss: 1.4434 - val_acc: 0.7450
Epoch 4/15
6000/6000 [==============================] - 4045s 674ms/step - loss: 0.0126 - acc: 0.9961 - val_loss: 1.5555 - val_acc: 0.7500
Epoch 5/15
6000/6000 [==============================] - 4044s 674ms/step - loss: 0.0103 - acc: 0.9967 - val_loss: 2.0543 - val_acc: 0.7150
Epoch 6/15
6000/6000 [==============================] - 4050s 675ms/step - loss: 0.0092 - acc: 0.9973 - val_loss: 1.9973 - val_acc: 0.7250
Epoch 7/15
6000/6000 [==============================] - 4048s 675ms/step - loss: 0.0082 - acc: 0.9976 - val_loss: 1.8895 - val_acc: 0.7550
Epoch 8/15
6000/6000 [==============================] - 4055s 676ms/step - loss: 0.0077 - acc: 0.9979 - val_loss: 2.2594 - val_acc: 0.7050
Epoch 9/15
6000/6000 [==============================] - 4040s 673ms/step - loss: 0.0065 - acc: 0.9981 - val_loss: 2.0215 - val_acc: 0.7050
Epoch 10/15
6000/6000 [==============================] - 4049s 675ms/step - loss: 0.0063 - acc: 0.9983 - val_loss: 2.4165 - val_acc: 0.7100
Epoch 11/15
6000/6000 [==============================] - 4045s 674ms/step - loss: 0.0059 - acc: 0.9985 - val_loss: 2.0839 - val_acc: 0.7650
Epoch 12/15
6000/6000 [==============================] - 4054s 676ms/step - loss: 0.0050 - acc: 0.9987 - val_loss: 2.0945 - val_acc: 0.7750
Epoch 13/15
6000/6000 [==============================] - 4246s 708ms/step - loss: 0.0058 - acc: 0.9985 - val_loss: 2.4129 - val_acc: 0.7150
Epoch 14/15
6000/6000 [==============================] - 3950s 658ms/step - loss: 0.0056 - acc: 0.9986 - val_loss: 2.3550 - val_acc: 0.7600
Epoch 15/15
6000/6000 [==============================] - 4263s 711ms/step - loss: 0.0046 - acc: 0.9988 - val_loss: 2.7727 - val_acc: 0.7200
Out[1]: <keras.callbacks.History at 0xf144aca128>
```

After completing the execution of the program the conclusive results are shown in table5.1

Table 5.1 Validation and test results of image detection

S.No	Output Parameter	Validation	Test
1	Accuracy	0.72	0.9988
2	Loss	2.77	0.0046

By above table our conclusion is as follows

- We choose a sample size of 2000 images
- We choose 200 images for testing and also for validation
- We choose 15 epoch and choose 6000 iteration in each epochs.
- We got the validation accuracy is 72%
- We got the Test accuracy is 99.88%

Here test accuracy is much higher than the validation accuracy. Test accuracy is 99.88 % which reflect the higher degree of precision of network. It may be because of the over fitting of the network. But finally we our designed network is giving the 99.88% prediction result and it clearly classify the difference between

VII. CONCLUSION

Test criteria for image detection are average loss and accuracy. Here we are also calculating the validation loss and validation accuracy. For input layers we choose the activation function RELU and for output layer we choose the SIGMOID as activation function. We have 'ADAM' is the optimization algorithm and the loss function is 'binary cross entropy. After completing the execution of the program we found that test accuracy of the network is 99.88% and the validation accuracy is 72%. The output result is approximately 100% correct, which implies that our network is over fitted. But if we choose the same network for same application it will work with same accuracy.

REFERENCES

1. *R. S. Cabral, F. Torre, J. P. Costeira, and A. Bernardino. Matrix completion for multi-label image classification. In Advances in Neural Information Processing Systems, pages 190–198, 2011.*
2. *T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In Proceedings of the ACM international conference on image and video retrieval, page 48. ACM, 2009. 5,*
3. *M. M. Cisse, N. Usunier, T. Artieres, and P. Gallinari. Robust bloom filters for large multilabel classification tasks. In Advances in Neural Information Processing Systems, pages 1851–1859, 2013.*
4. *G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 8609–8613. IEEE, 2013.*
5. *J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. IEEE, 2009.*
6. *M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303– 338, 2010.*
7. *A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In Advances in Neural Information Processing Systems, pages 2121–2129, 2013.*
8. *N. Ghamrawi and A. McCallum. Collective multi-label classification. In Proceedings of the 14th ACM international conference on Information and knowledge management, pages 195–200. ACM, 2005.*
9. *Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. arXiv preprint arXiv:1312.4894, 2013.*