# Cat Vs Dog Image Classification : Effect of Dropout on Model Performance

Swati Banthiya
Student at College of Engineering &
Computer Science
Florida Atlantic University
Boca Raton, Florida

***Abstract*—This paper summarizes results of an experimental study carried out to understand the effect of different number of dropout layers and degrees of dropout on the performance of a base model. The base model is based on a Convolutional Neural Network. The Cat Vs Dog images dataset was borrowed from the Kaggle database.**

***Keywords*—CNN, Dropout**

## I. INTRODUCTION

Convolutional Neural Networks (CNN) are preferred over dense multilayer networks for image processing because of their ability to find local features rather than global features. Images are also generally larger in size and number, making it computationally impossible to use dense networks. Although CNN will reduce parameters number multifold by identifying the local patterns, this binocular vision of CNN could result in the model to overfit. Overfit models may not perform well on the new datasets and hence the need of model regularization. Dropout also makes it possible to use the different large networks developed during training to be used for testing, which is they have been referred to as a technique of sampling of neural networks by Srivastava et al., 2012 [1]. The research problem identified in this report is the impact of number of dropout layers and degrees of dropout on the validation accuracy of a model. The methodology adopted to conduct the study is to develop a base model, followed by changes to dropout layer and dropout parameters to develop five models. The expected behavior of the five models will be compared to the actual behavior. Using the learnings from the five model, a final 'Improved model' will be suggested. The scope of the study is limited to developing five new models.

## II. METHODOLOGY

### A. *Base Model: Benchmark*

The first step was to design a base model. The base model was not designed to achieve a high level of accuracy, but to act as a base for further work and provide ease of computational speed.

Four thousand images were selected from the Kaggle database and were split into training dataset, validation dataset and test dataset in the ratio 2:1:1 with equal number of dog and cat images.

The base model designed for the project comprises the following layers.

Three convolutional layers: The role of the convolutional layer is to create feature maps to predict the class probabilities

Two max pooling layers: The role of the max pooling layer is to identify the more prominent feature in an image.

Two dense layers: Dense layers have every node in one layer connected to every node of the next layer, helping the network learn the global patterns.

One dropout layer: The dropout layer temporarily disabling connection to and from a neuron. Nothing is deleted; hence the architecture of the network remains unchanged.

### B. *Learning Methods and Goals*

New models created were a variation of the base model. One factor at a time was tweaked to develop a new model so that any improvement or depletion in accuracy can be associated with the change. All the models used the base model and so all the changes in accuracy are calculated with reference to baseline accuracy.

The learning of the five models is then put together to yield the best improvement in accuracy. This is named as the 'Improved model' and is placed next to the 'base model' to display the overall product of the experimental study.

The Network-in-Network, 2013 study [3] has changes in loss ranging between 1% to 7% compared to the baseline as per Table 1 on the CIFAR-10 data. Park and Kwak, 2016 [2], reduced classification error by 15% on the CIFAR-10 data. Although these changes cannot be directly compared and are based on the dataset, parameters being used, the scope of the project and the starting base accuracy, a goal of at least 15% improvement in accuracy is set at the start of the project.

## III. EXPERIMENTS

The parameters of the Base model are provided in the diagram below in Table 1.

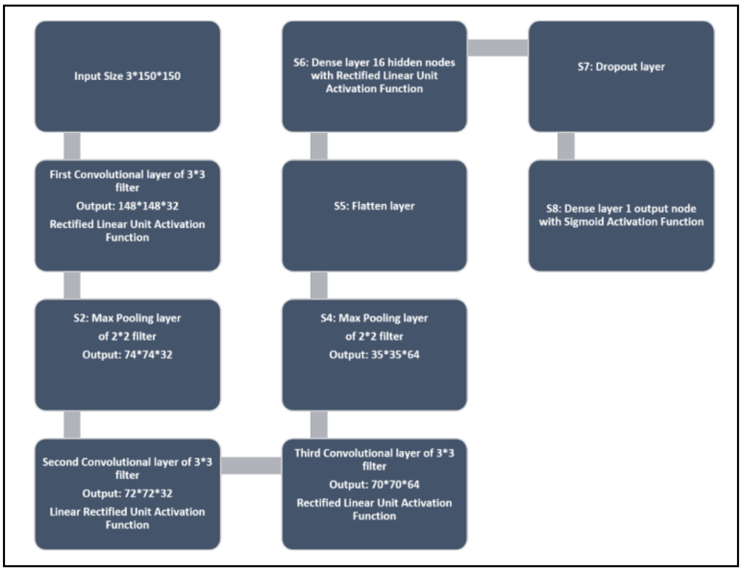


Figure 1: Experimental Settings of Base Model

## A. *Models and their Expected Behavior*

The different models and what they are expected to do in terms of performance are described below.

Model #1: Add a dropout layer after the second convolutional layer in Table 1 with a dropout rate of 0.5. This change is expected to improve the performance of the base model as Park and Kwak, 2016, [2] describe in their paper, adding dropout to convolutional layers increases the regularization effect of dropouts when training sample size is small and data augmentation is not used, as is in the case of our models.

Model #2: Add a dropout layer after the second max pool layer (S4 in Table 1) with a dropout rate of 0.5. Regularization added to model is expected to improve base model performance.

Model #3: Change dropout rate in the dropout layer (S7 in Table 1) of the base model from 0.5 to 0.1. This model is expected to deplete the performance of the base model. Lower dropout rates resulted in high training error in Srivastava et al., 2012 [1] MNIST study.

Model #4: Add a dropout layer after the second max pool layer (S4 in Table 1) with a dropout rate of 0.1. Regularization added to model is expected to improve base model performance but reducing dropout rate may deplete the performance. Hence, the improvement in model #2's performance is expected to be higher than for model #4.

Model #5: This model uses spatial dropout. As described by Brownlee J. (2018), spatial dropout will drop all the features maps from the convolutional network. Spatial dropout showed different results on CIFAR and MNIST data as per the study conducted by Park and Kwak, 2016, [2]. Hence the expected result is unclear.

## B. *Actual Model Performance*

A summary of the performance of the above models is provided below.

| Model ID | Test Accuracy | Change in Accuracy |
|---|---|---|
| **Base Model** | 50.00% | 0 |
| **Model #1** | 64.30% | 28.60% |
| **Model #2** | 63.80% | 27.60% |
| **Model #3** | 66.40% | 32.80% |
| **Model #4** | 56.80% | 13.60% |
| **Model #5** | 50.00% | 0 |

Figure 2: Model Performance Summary

Comparing the expected versus the actual performance of the models with respect to the base model, all the models behaved as expected, except for Model 3, where the change in dropout resulted in increase in accuracy instead of a decrease in accuracy.

## C. *Base Model Vs Improved Model*

To design the final 'Improved model', we begin by incorporating all the changes made to the models that resulted in a performance improvement of more than 15%. This included model 1(adding a dropout layer after the second convolutional layer), model 2(adding a dropout layer after the second max pool S4) and model 3 (reducing the dropout rate in S7 from 0.5 to 0.1).

Combining models 1, 2 and 3's changes did not result in any improvement an accuracy. So the next step was to combine the changes of model 1 and 3 only. This resulted in a model with an accuracy of 68% and was defined as the 'Improved model'.

The Improved model showed an improvement in accuracy of 36% compared to the base model.

The following figures show the comparison of the Base model to the Improved model in terms of epoch vs the accuracy of the training and the testing datasets. The fact that they are both closely knitted allows us to conclude that there is no overfitting.

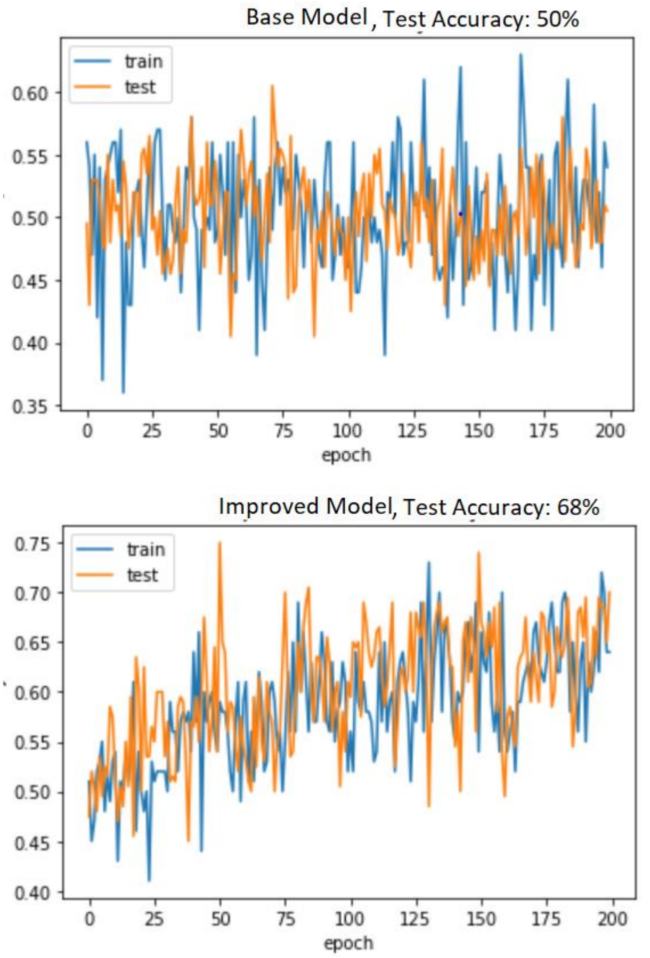


Figure 3: Base Model Vs Improved Model Performance

## IV. CONCLUSION

Dropout layers and rates and their impact on convolutional neural networks can be very tricky to deal with because the results can change every time the model is run and to which layer it is being applied to. The experimental study reveals that adding dropout layers after the convolutional layer and decreasing the dropout rate of dropout layer after the first dense layer resulted in the maximum improvement in test accuracy.

## ACKNOWLEDGMENT

All programming tasks performed for this project were carried out by modifying Professor Xingquan Zhu base code. The Colab Notebook is available on Github - [https://github.com/SBANTHIYA/ASSIGNMENT2-SBANTHIYA/blob/master/FinalDropoutProject.ipynb](https://github.com/SBANTHIYA/ASSIGNMENT2-SBANTHIYA/blob/master/FinalDropoutProject.ipynb)

## References


[1] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, 1929–1958.

[2] Park, S., & Kwak, N. (2016). Analysis on the Dropout Effect in Convolutional Neural Networks. *ACCV*.

[3] Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)

[4] Wu, H., Gu, X.: Towards dropout training for convolutional neural networks. Neural Networks 71 (2015) 1–10

[5] Brownlee, J. (2018). How to Reduce Overfitting With Dropout Regularization in Keras retrieved from https://machinelearningmastery.com/how-to-reduce-overfitting-with-dropout-regularization-in-keras/