

## Assignment 3 – Cats Vs Dogs

Vineeth Goud Maddi

### Analysis:

**Task:** Given an image, our task is to predict whether an image is Cat or Dog.

**Dataset:** I downloaded the dataset from the Kaggle, it is an official dataset and has training samples of 25000 images, test, and validation of 5000 images each.

As per the instructions provided in the assignment, I initially started training the data with 1000 samples of train, 500 test and validation each. No optimizations, it's just a basic convnet from scratch.

And the results are displayed in the table.

In the second case, I optimized the network with dropouts and performed the augmentation.

In the later stage, I added more training samples and repeated the same process, case with optimization and augmentation.

For Optimization, the methods below can help us to solve overfitting problems.

- **Gaining access to more training data.**
- **Data augmentation**
- Cross validation
- **Addition of noise to the input data**
- **Making the network simple or tuning the capacity of the network** (more capacity than required leads to a higher chance of overfitting).
- Regularization.
- **Adding dropouts**

The highlighted things are used in my network.

Also, I used ResNet152V2 architecture, and the code referred from the link

(<https://keras.io/api/applications/resnet/#resnet152v2-function>).

ResNet152V2 is an improved version of the ResNet152 architecture introduced in 2016 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun from Microsoft Research. ResNet152V2 is designed to improve the accuracy and efficiency of the ResNet152 architecture while maintaining its depth and structure.

The architecture of ResNet152V2 is similar to ResNet152, but with a few key differences:

Pre-activation: In ResNet152V2, the batch normalization and ReLU activation are applied before the convolutional layer instead of after, as in the original ResNet. This "pre-activation" approach has been shown to improve the accuracy and speed of training deep networks.

Bottleneck blocks: ResNet152V2 uses bottleneck blocks, which consist of three convolutional layers with different filter sizes (1x1, 3x3, and 1x1) to reduce the number of parameters and improve efficiency.

Grouped convolutions: ResNet152V2 uses grouped convolutions in some of its layers, which splits the input into groups and applies a separate convolutional filter to each group. This can reduce the computational cost of the network and improve its efficiency.

(The above content referred from the internet)

I categorize my analysis in following:

**Let's now compare the cases for the network that is created from scratch.**

Cases	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Test accuracy	Observations
Basic Convnet from scratch (no dropout, data augmentation)	100	73.80	1.2	2.04	64.70	Clearly from the results we notice that our model works on training data and behaves poorly on validation and test data. (Overfitting problem)
Basic Convnet with data augmentation and dropouts	95.30	81.80	0.12	0.68	80.90	Here we notice that our model is consistent in with training and validation loss and accuracy. Though our training accuracy is reduced to 93. We eliminated some overfitting problem.
Convnet with more training samples, validation, and test samples	99.08	77.20	0.03	1.85	78.3	Here we can see that overfitting problem can be avoided by adding more data samples.
With Data Augmentation and dropout with more training, validation and test samples	84.93	86.30	0.4	0.46	83.3	In this case, it is weird that our results accuracy has dropped, and it is more consistent.
Optimizer changed to ADAM, dropouts and	99.37	89.47	0.11	0.32	89.34	This case has showed best performance results when we all the optimizations and regularizations techniques.

data augmentation, regularizer						
--------------------------------	--	--	--	--	--	--

I used **ResNet152V2** architecture as a pretrained network and below are the results.

Cases	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Test accuracy	Observations
Using ResNet152V2 as base	100	98.40	1.72	4.70	97.60	Overall, the result was decent using the ResNet152V2 and I noticed a validation loss is more which can be reduced with some optimizations
Using ResNet152V2 with data augmentation and dropouts	99.80	98.20	0.06	1.23	97.80	We noticed that accuracy has increased, and validation loss is reduced.
Using ResNet152V2 as base with more training, validation, and test	98.90	97.90	0.5	2.4	97.6	It also shows the decent result, but it has validation loss
With Data Augmentation & Dropouts	99.65	98.20	0.04	0.8	97.7	Best result so far with optimizations & Data Augmentation techniques.

### Conclusion:

Overall, the initial performance of the basic network is poor and then later we significantly improved the results with data augmentation, adding some random noise to the data. Later I tried to use some advanced optimizers additionally to see the results. With all the modifications, we improved accuracy and reduced loss.

Later I started the model with ResNet152V2 as base and trained the model, initial results were good, and we can notice that the accuracy is above 97.7 percent and it significantly improved with the other techniques.

Best performance plots are shown below,

