CONVOLUTION REPORT

Chakradhar Challa

To build a convolutional neural network by considering the example of cats and dogs using two approaches, namely training from scratch and using a pretrained convnet to reduce overfitting that leads to more difficult prediction during training, several techniques called data augmentation and regularization.

Data: I took the data of cats-vs-dogs as the approach from Kaggle and made a new dataset with training, validation, and testing samples:

- Training set with 1000 samples
- Validation set with 500 samples.
- Training set with 500 samples

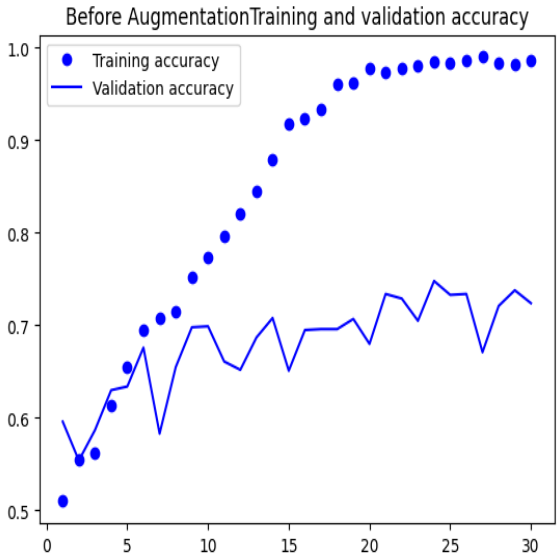
1. **Consider the Cats & Dogs example. Start initially with a training sample of 1000, a validation sample of 500, and a test sample of 500 (like in the text). Use any technique to reduce overfitting and improve performance in developing a network that you train from scratch. What performance did you achieve?**

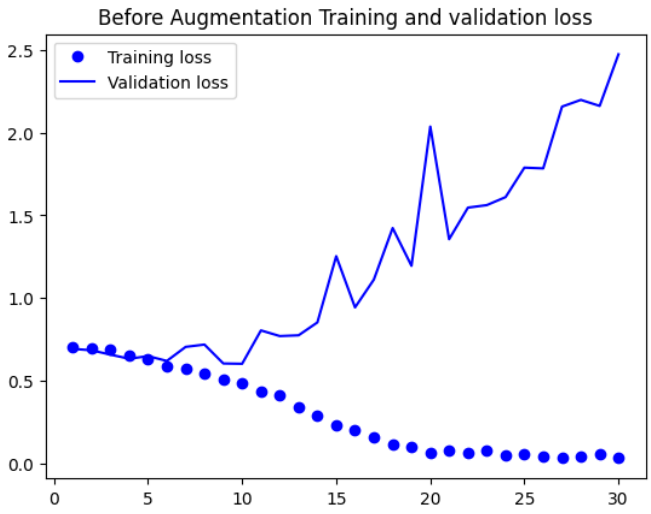
Data preprocessing (rescaling and eliminating unnecessary data)

- As per the approach, we need to train a network from scratch. From the above data, I have loaded the dataset and need to import the required modules, like the operating system interface, high-level file operations, object-oriented filesystem paths, etc., which are essential for working with directories and some other filesystems.

Building the model

Here, the input for this network is a 3-D tensor, which is images that need to be reshaped. For that, we need to use the general model structure for the convnet with alternated Conv2D (with 'relu' activation) and maxpooling2D stages more due to their bigger images.





From the above plots, the characteristics of overfitting are that the training accuracy increases linearly over time and nearly reaches 100%, whereas the validation accuracy is only 75%.

And the validation loss stalls up to 10 epochs after it steadily increases, whereas the training loss keeps decreasing linearly as training proceeds.

The validation and test accuracy (71%) of this model seems to be very low we need to reduce overfitting by using few techniques namely:

a. Data Augmentation
b. Dropout method
c. Using both

Model	**Validation Accuracy**	**Test Accuracy**
Sample training	0.74	0.71
Data Augmentation	0.8	0.81
Dropout	0.77	0.74
Using both (Data Augmentation & Dropout	0.82	0.83

- According to the performance metrics of validation and test of all the models I compared only the model with Data augmentation and Dropout gives the best performance above all techniques. I.e. it is the best technique and to use remaining samples to regularize the model.

2. **Increase your training sample size. You may pick any amount. Keep the validation and test samples the same as above. Optimize your network (again training from scratch). What performance did you achieve?**

- Here in this model, we increased the training sample size to 7000 to improve the accuracy of the model and that one technique that can increase the training sample within the data is Data Augmentation and Dropout method of which we get better performance metrics, and the performance has been evaluated.

Model	**Test loss**	**Validation Accuracy**	**Test Accuracy**
Data Augmentation & Dropout method	0.29	0.93	0.9

From the above model performance metrics, the validation and test accuracy are increased when we increase our training sample by 10% that's because we implemented the technique of Data augmentation and dropout method.

3. **Now change your training sample so that you achieve better performance than those from Steps 1 and 2. This sample size may be larger, or smaller than those in the previous steps. The objective is to find the ideal training sample size to get the best prediction results.**

- As per the approach we need to increase the training sample size by 10000 and further performance was observed because we do not know the appropriate sample size to further enhance the model performance.

Training Sample	**Validation Accuracy**	**Test Accuracy**	**Test Loss**
1000	0.74	0.71	0.55
8000	0.9	0.88	0.29
10000	0.91	0.9	0.25

- From the above table, the training sample with 10000 has high validation accuracy and test accuracy whereas test loss was very low with technique called data augmentation.
- This technique is better compared to the previous model which is unregularized.

4. **Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use all optimization techniques to get best performance.**

- Using Pretrained model—This model is previously trained on a large dataset, that in turns gives powerful features/representations within the data and acts as a generic model that will be useful for computer vision problems like we can train a model on ImageNet (where classes are mostly animals and everyday objects). This network is called VGG16, which is a simple design for ImageNet.

Feature Extraction with data augmentation

Pretrained Model	Validation Accuracy	Test Accuracy	Test Loss
Feature Extraction	0.97	0.97	0.22
Fine Tuning	0.97	0.95	0.088

The above pretrained models are done with data augmentation where we get better performance metrics that will be helpful for further analysis.

Pretrained Model with samples	Validation Accuracy	Test Accuracy	Test Loss
1000	0.97	0.95	0.18
8000	0.98	0.98	0.1
10000	0.98	0.97	0.12

- We got better performance metrics for the pretrained models with fine tuning while increasing samples size as per the approach.
- Since, we used pretrained features that already have knowledge about cats and dogs.

Conclusion—

- The training accuracy of an unregularized model was 91% because we use data augmentation and dropout as a technique that will reduce overfitting.

- With the help of convnets we trained the model from scratch even though we have a small dataset that can be learned a hierarchy of modular patterns and concepts that represent visual world.
- The main problem is that in this model was overfitting and to overcome this, we need to implement several techniques, in that we have data augmentation and dropout, a powerful way to fight against overfitting when we are working with this type of data.
- As we used feature extraction and, we need to use fine-tuning that adapts to a new problem will push the performance a bit further like we see the training accuracy and validation accuracy are both increased due to it fits the new model.
- By increasing the training sample we can overcome the overfitting with data augmentation and dropout that leads to better performance metrics.