Cats Vs Dogs Classification

Shweta Srinivasan

OBJECTIVE

- This kaggle competition project aims to classify cats and dogs through image classification.
- The training data comprises of 25000 images and the test data consists of 1000 Images.
- The submissions made on kaggle are on the test data.
- The goal is to develop a classification technique in the machine learning landscape, particularly in deep learning.

VARIOUS APPROACHES

- Approach 1, used the MNIST classification as a reference and built a basic Convolution Neural Network using Keras with Tensorflow as a back-end. The initial pass produced a log loss of 8 approximately and accuracy of the order of 50% .
- As a second step to approach 1, the basic CNN architecture was modified by adding few more Convolution 2D layers, maxpooling layers and dense layers.
- Also experimented with dropout rates modifying it between 0.2 and 0.8. The results of this model did not show significant improvement to the accuracy and log loss error, despite increasing number of epochs.
- The next approach I made was to attempt the problem fully from a Tensorflow perspective.
- I started of with an architecture similar to that of MNIST using Tensorflow, the results of which were quite satisfactory.

DESIGN

- The architecture I chose ultimately gave me the best results which was effective from a model run time perspective as well.
- The architecture builds on MNIST and consists of 6 convolution layers, 1 fully connected layer.
- The activation function used being softmax, adam optimizer and a learning rate of 1e-3.
- The further I decreased the learning rate, the model took longer to train with not much improvement in accuracy and hence I settled with this particular learning rate.

IMPLEMENTATION

- Preprocessing includes one hot encode label values for cat and dog, cat → [1,0]; dog→ [0,1]
- As the next step in preprocessing, the image is converted to grayscale and converted to a uniform size of 50x50 for uniformity.
- The next step is to create the train and test data by appending each image with the corresponding class label, train data also shuffled.
- The model architecture is defined in the next stage, with the suitable (6) convolution layers, 1 fully connected layer.
- The train data is split and 500 images are retained for testing model accuracy.
- Since the training data now consists of images with the labels, the data needs to be split to retrieve just the images.
- Following this, the images are reshaped before feeding into the neural network.
- The final step is to fit the model and observe results for 4 epochs.

RESULTS

- The model achieves an accuracy of 84.05% with a log loss of 0.35 and an accuracy of 78.80 and log loss of 0.47 on the validation set.
- The log loss is calculated as:

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right],$$

where

- n is the number of images in the test set
- *yî* is the predicted probability of the image being a cat
- yi is 1 if the image is a cat, 0 if dog
- *log*() is the natural (base e) logarithm

FINDINGS

- This architecture works well to classify 25000 images and does a satisfactory job on validating the model against the test set as well.
- The modification of learning rate and addition of convolution layers was key to increasing performance for this particular problem statement.
- A smaller learning rate took slightly longer but covered the possibility of discovering the best fit as opposed to using a larger learning rate which might have the tendency to miss the best fit.

AREAS OF IMPROVEMENT

- This architecture was built on the MNIST architecture and hence could always be improved further by tweaking learning rate, number of layers, dropout rate.
- The Keras implementation could also be tweaked to improve accuracy, by modifying the tuning parameters.
- One aspect which I could do differently would be to completely redesign this architecture to mimic an architecture like Alex Net and try and replicate the results.

CONCLUSION

- The goal to classify cats and dogs was successfully implemented using a neural network which does a good job of classifying images with an accuracy of 83%
- I would definitely recommend this model to classify images of a similar nature for any future studies.

QUESTIONS?