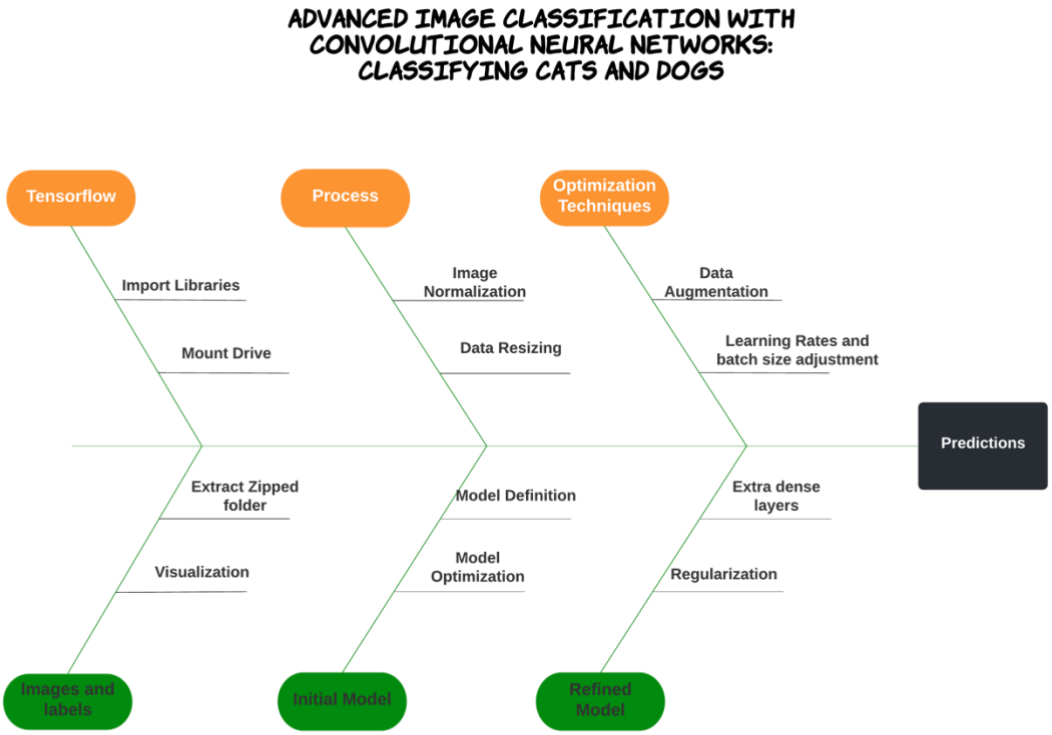
Advanced Image Classification with Convolutional Neural Networks: Cat or Dog?

A Critical Review

Introduction

This article reviews a convolutional neural network designed to classify cats and dogs with the objective of achieving accurate classification.



Methodology:

The initial neural network architecture in TensorFlow and Keras consisted of an input layer, 3 convolutional and pooling layer pairs, and an output layer. A range of advanced techniques were employed including Data Augmentation, varying learning rates, batch sizes, batch normalization, and dropout layers. The models were tested with different sets of hyperparameters, which are summarized below.

Summary

Model	Number of Layers	Features	Number of Epochs	Data Augmentation	Validation Accuracy	Validataion Loss
M1	3: 16, 32, 512 units	Relu Activation Function,	5	No	0.516	0.3549
M2	5: 32, 64, 128, 256, 512 units	Relu Activation Function	50	Yes	0.7612	2.0717
M3	5: 32, 64, 128, 256, 512 units	Relu Activation Function, Dropout: 0.2	50	Yes	0.7798	2.069
M4	5: 32, 64, 128, 256, 512 units	Relu Activation Function, Dropout:0.5, Data Augmentation	50	Yes	0.8990	0.2577
M5	5: 32, 64, 128, 256, 512 units	Relu Activation Function, Dropout:0.5, Data Augmentation, Batch Normalization	50	Yes	0.9051	0.2462

Results

i) The model's final accuracy was 0.9051 with an error rate of 0.2264, visualized using Plotly.

ii) Improvements to the model's accuracy were achieved by adding two pairs of convolutional and pooling layers, resulting in an accuracy of 0.7612.

iii) A dropout rate of 0.2 further improved accuracy to 0.7798 while increasing the dropout rate to 0.5, with data augmentation increased accuracy to 0.8990.

iv) Adding batch normalization significantly improved accuracy, resulting in a final accuracy of 0.9051 and a validation loss of 0.2462.

Snapshots

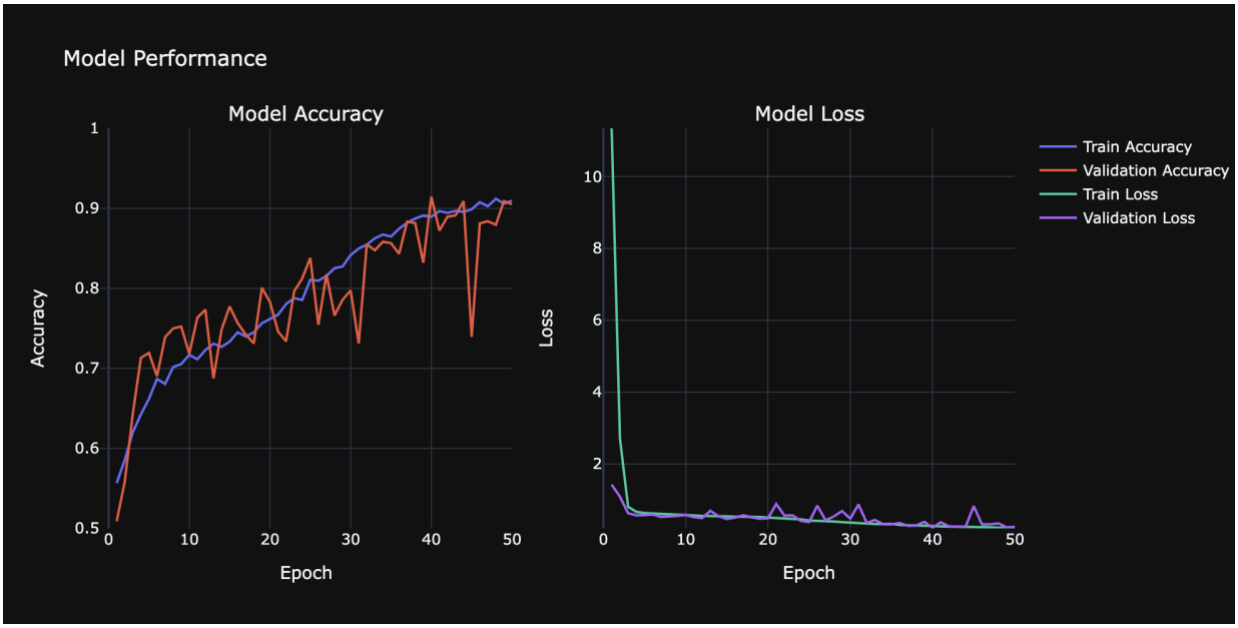
A. Model Summary

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 298, 298, 32)      896

max_pooling2d_4 (MaxPooling  (None, 149, 149, 32)      0
2D)

batch_normalization_4 (Batc  (None, 149, 149, 32)      128
hNormalization)

conv2d_5 (Conv2D)            (None, 147, 147, 64)      18496

max_pooling2d_5 (MaxPooling  (None, 73, 73, 64)        0
2D)

batch_normalization_5 (Batc  (None, 73, 73, 64)        256
hNormalization)

conv2d_6 (Conv2D)            (None, 71, 71, 128)       73856

max_pooling2d_6 (MaxPooling  (None, 35, 35, 128)       0
2D)

batch_normalization_6 (Batc  (None, 35, 35, 128)       512
hNormalization)

conv2d_7 (Conv2D)            (None, 33, 33, 256)       295168

max_pooling2d_7 (MaxPooling  (None, 16, 16, 256)       0
2D)

batch_normalization_7 (Batc  (None, 16, 16, 256)       1024
hNormalization)

flatten_1 (Flatten)          (None, 65536)             0

dense_2 (Dense)              (None, 512)               33554944
```

B. Final Model Accuracy:

```
187/187 [==============================] - 146s 779ms/step - loss: 0.2498 - accuracy: 0.8988 - val_loss: 0.8198 - val_accuracy: 0.739
Epoch 46/50
187/187 [==============================] - 149s 796ms/step - loss: 0.2256 - accuracy: 0.9077 - val_loss: 0.3116 - val_accuracy: 0.8811
Epoch 47/50
187/187 [==============================] - 147s 787ms/step - loss: 0.2320 - accuracy: 0.9026 - val_loss: 0.3138 - val_accuracy: 0.8839
Epoch 48/50
187/187 [==============================] - 146s 779ms/step - loss: 0.2194 - accuracy: 0.9120 - val_loss: 0.3382 - val_accuracy: 0.8795
Epoch 49/50
187/187 [==============================] - 147s 786ms/step - loss: 0.2301 - accuracy: 0.9058 - val_loss: 0.2289 - val_accuracy: 0.9090
Epoch 50/50
187/187 [==============================] - 149s 795ms/step - loss: 0.2195 - accuracy: 0.9097 - val_loss: 0.2462 - val_accuracy: 0.9051
```

C. Accuracy and Error plots:



Review:

i) Batch Normalization improved accuracy by standardizing the activations of each layer to have zero mean and unit variance, leading to faster convergence, generalization, and increased robustness to hyperparameters.

ii) Increasing the Dropout rate from 0.2 to 0.5 decreased training accuracy but improved validation accuracy, suggesting enhanced generalization to novel data.

iii) The oscillations in the validation curve can be attributed to the separate data used for validation, while similar increase in training and validation accuracies indicated no overfitting.

iv) The comparable decrease in training and validation errors indicated no overfitting, while the zig-zags in the validation error curve could be attributed to the use of separate data for validation.

v) Data Augmentation also played a key role in accuracy by exposing the model to varied distortions of same images, thus training it to predict more accurately.