InceptionV3 Architecture for Flower Classification

Bishwas Mishra, Furkan Ozbudak, Muhammad Luqman, Jirom Hiyaba Mebrahtu, and Taha Abdelrahman Mahmoud Metwally

Machine Learning - July 2020, Maharishi International University, Fairfield, Iowa, USA

Abstract

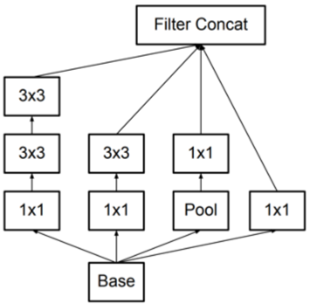
The pre-trained Inception-v3 model achieves state-of-the-art accuracy for recognizing general objects with 1000 classes, like "Zebra", "Dalmatian", and "Dishwasher". The model extracts general features from input images in the first part and classifies them based on those features in the second part [1]. In this paper we have used InceptionV3 model based on transfer learning to classify publicly available dataset: Flowers[2]. We achieved a performance of 96.49% accuracy based on our hyperparameter tuning.

Keywords: InceptionV3, Flowers Recognition, CNN

1. Introduction

Transfer Learning is used as, for any given domain D_S and learning task T_S, and another domain D_T and learning task T_T, it helps improve the learning of the target predictive function f_T in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$.[3]

Inception v3 is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. It consists of factorization of 5X5 convolution into 3X3 convolution because large filters are expensive in terms of computation.



The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax. [3]

2. Proposed Model

We input the Flowers images into the feature extraction part of Inception-v3 which converts the 2662 images data into feature vectors consisting of 2048 float values for each image.

The next step is to train the classification part of the model using the preprocessed data.

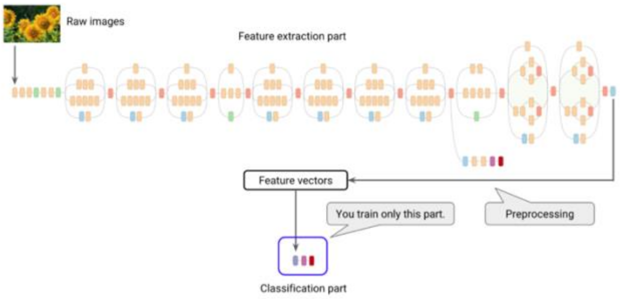


Image: 2 [4]

3. Test Results

3.1 Experiment - 1

The first experiment we did with the validation data is to identify how much does higher input resolution helps if the computational effort is kept constant.

Image Size, batch, epoc	Accuracy
151 X 151, 32, 10	85.66%
299 X 299, 32, 10	96.69%
399 X 399, 32, 10	(very time consuming)

Then we checked for the model with some data augmentations and hyperparameter tuning.

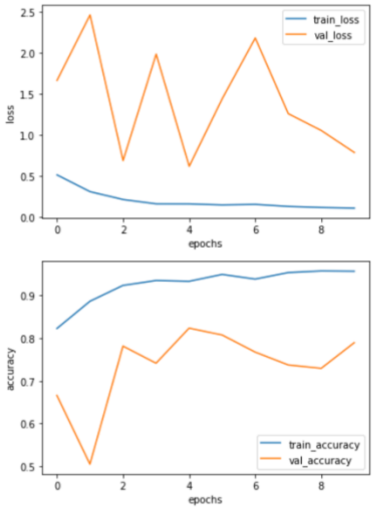
3.2 Experiment - 2

Here,

I. image size is 299 X 299
II. batch size=32,
III. EarlyStopping
    a. monitor='val_loss',
    b. min_delta=0.001,
    c. patience=10,
    d. verbose=1,
    e. mode='auto'
IV. ReduceLROnPlateau
    a. monitor='val_loss',
    b. factor=0.1,
    c. patience=3,
    d. verbose=1,
    e. mode='auto'
V. ModelCheckpoint.

```
image_datagen = ImageDataGenerator(
    rescale=1./255,
    vertical_flip = True,
    horizontal_flip = True,
    rotation_range=20,
    shear_range=0.05,
    zoom_range=0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    validation_split=0.2
)
```

Accuracy: 95.69%

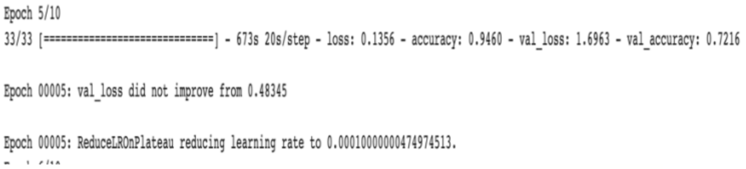


There is clearly overfitting because the training accuracy is not similar to the validation accuracy.

3.3 Experiment – 3

Here,

I. batch size=64
II. SAME other parameters



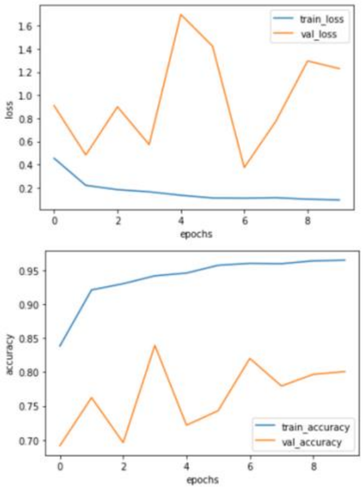
```
Epoch 5/10
33/33 [==============================] - 673s 20s/step - loss: 0.1356 - accuracy: 0.9460 - val_loss: 1.6963 - val_accuracy: 0.7216

Epoch 00005: val_loss did not improve from 0.48345

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
```

```
image_datagen = ImageDataGenerator(
    # SAME AS PREVIOUS +
    channel_shift_range=0.1
)
```

Accuracy: 96.49%



5. Conclusions

There is underfitting problem due to small dataset. We need to augment the data more to have better results. The best result we got was 96.49% for experiment 3. This can be imprived further regarding overfitting.

Acknowledgements

We would like to thank Professor Antony Sanders for giving us this opportunity for hands on with CNN and Deep Learning. We also thank him a comprehensive introduction to this limitless subject domain called Machine Learning.

References

[1] https://codelabs.developers.google.com/codelabs/cpb102-txf-learning/index.html#1

[2] Flowers Dataset, https://www.kaggle.com/alxmamaev/flowers-recognition

[3] https://en.wikipedia.org/wiki/Transfer_learning

[4] https://cloud.google.com/tpu/docs/inception-v3-advanced