2019/10/16
0780828
Alisher Mukashev (穆安里)

HOMEWORK #1

Goals

Train your convolution neural network to classify the natural scene images into correct class.

Introduction

The homework is related to image classification problem. This problem can be solved using deep learning models. The dataset consists of 3859 gray images belonging to 13 classes (train: 2819, test: 1040). Example of a train image is shown on Figure 1:

Figure 1: Class “bedroom”

Deep learning model must be trained and be able to recognize all classes. For better results, transfer learning and ensemble learning was chosen.

Methodology

Data pre-processing

Given dataset has unbalanced number of images. For example, class "*bedroom*" includes 136 images, while class "*opencountry*" includes 330 images. Hence, the number of images in the classes was increased by copying existing data to make this number equal for each class.

The function "*preprocess_input*" provided by Keras [1] was used to normalize training and test data.

Data Augmentation

The class "*ImageDataGenerator*" provided by Keras generates batches of tensor image data with real-time data augmentation. The data will be looped over (in batches). For given homework only argument "*horizontal_flip*" was set because it is the most suitable for that kind of dataset.

Model Architecture

Although the increasing of images was done, given dataset is still small. Hence, for better results transfer learning using VGG16, VGG19 [2] and ResNet50 [3] was chosen. In transfer learning, the pre-trained weights of an already trained model (one that has been trained on millions of images belonging to 1000's of classes such "*Imagenet*" dataset, on several high power GPU's for several days) are trained and these already learned features to predict new classes are used. Architectures of these transfer learning model, Figure 2.

Base models must be "frozen", i.e. turned off and only added layers should be trained. All added layers have a "*relu*" activation function and only last one is Softmax layer. The various dimensionalities of layers were tested and finally, the architecture that shown in Figure 3 was chosen. The dimensionality of the last Softmax layer is equal to the number of classes.

After that, to improve the results, the ensemble learning using all these deep learning models was applied. Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem. Ensemble learning is primarily used to improve the (classification, prediction, function approximation, etc.) performance of a model, or reduce the likelihood of an unfortunate selection of a poor one. Overall architecture is shown on Figure 3. Different dropout rates and different base models were used to compose overall model. As the ensemble technique the taking mode of the results is chosen. The mode is a statistical term that refers to the most frequently occurring number found in a set of numbers. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a separate vote. The prediction which we get from the majority of the models is used as the final prediction.

Hyperparameters

As an optimizer for all models "Adam" optimizer was chosen. The learning rate was selected by default (0.001). The batchsizes 32 and 64 were tested and finally 64 was chosen. The dropout rate in the range (0.3:0.5) in increments 0.5 were checked and finally 0.35, 0.4 were selected, as shown in Figure 3. The number of epochs in the range (30:60) were checked and the best results were shown when choosing 40. The input image size in the range (100:400) in increments 50 were tested and finally, 300x300 size were selected. All base models which were used in this work require RGB image as input, hence, the input image size dimension for training and testing process has become (300x300x3).

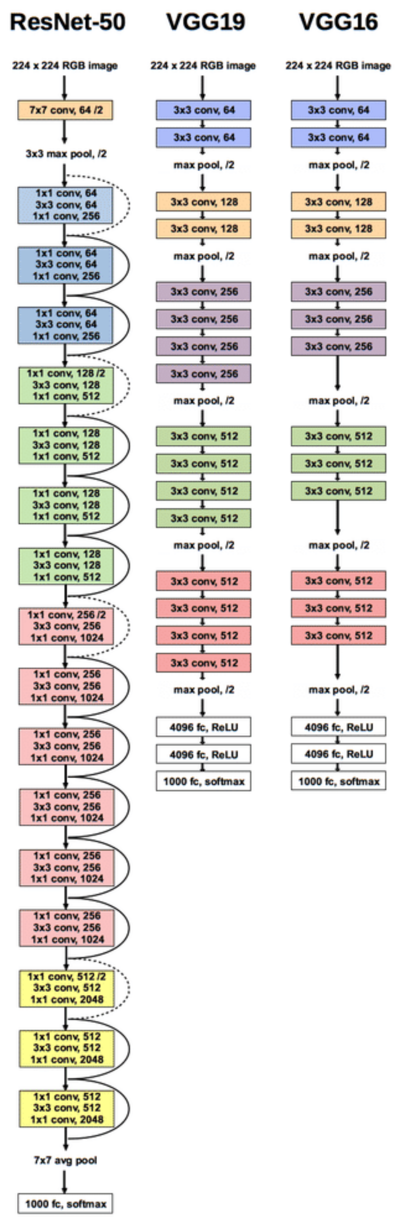

Figure 2: Base models

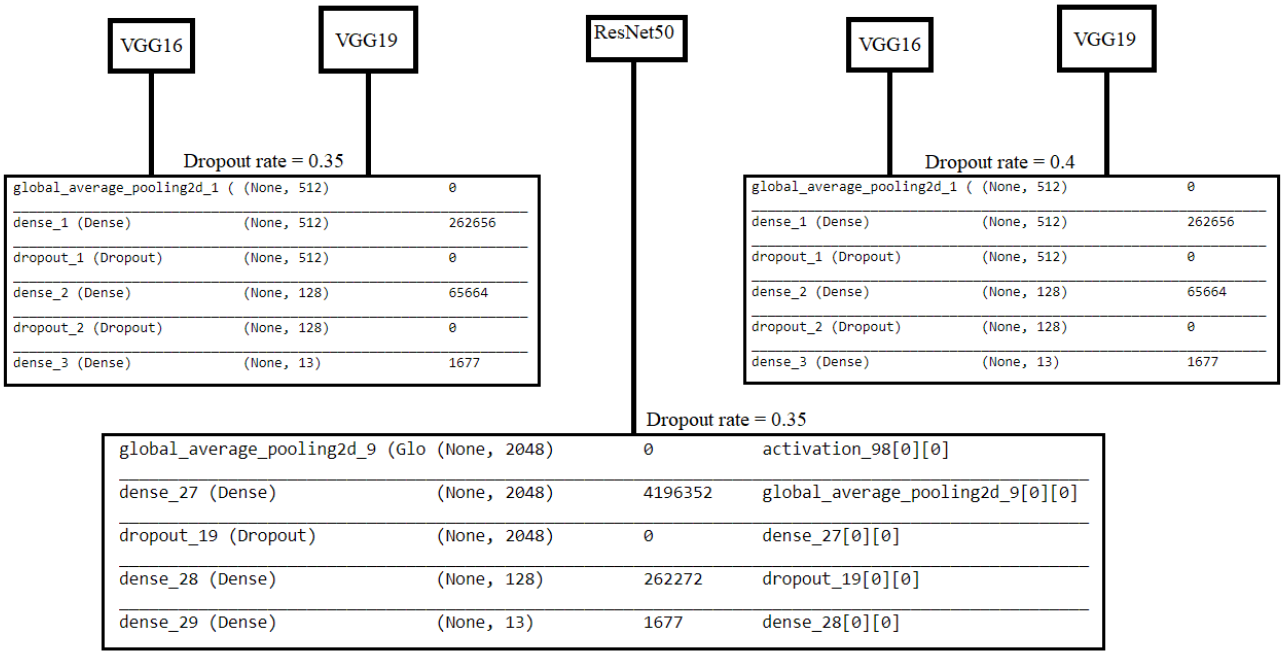

Figure 3: Model architecture

Finite hyperparamaretes:

Input image dimension – 300x300x3;

Batchsize – 64;

Optimizer – Adam;

Learning rate – 0.001;

Number of epochs – 40;

Dropout rate – 0,35 (VGG16, VGG19, ResNet50); 0.4 (VGG16, VGG19).

Avoid overfitting

To avoid overfitting, a validation set was used. The validation set was created by splitting training data by 85% and 15%. This training process using validation dataset made it possible to control loss parameter and choose the hyperparamateres.

After selecting the hyperparameters, 100% of the training data was used without any division to get the model.

Summary

Obtained model did not reach the rank top 3 in the leaderboard [4]. Ensemble learning model is very simple, there must be something more complex, like Adaptive Boosting.

References

1. https://keras.io/models/sequential/
2. https://arxiv.org/pdf/1409.1556.pdf
3. https://arxiv.org/pdf/1409.1556.pdf
4. https://github.com/alishsuper/Selected-Topics-in-Visual-Recognition-using-Deep-Learning/blob/master/HomeWork1.py