

A simple CNN to classify CIFAR-10 dataset based on VGG Architecture

Frank Acquaye

October 15, 2020

Abstract

This document describes an architectural design in solving the classification challenge of CIFAR-10. The proposed model achieves an accuracy above 0.77 on validation data-set.

1 Model Description

Owing to the understandable simplicity of the VGG architecture, this architecture was adopted in an attempt to solve the CIFAR-10 classification challenge.

The chosen architecture comprised primarily of a VGG Block. A VGG block consists of:

- A Convolutional layer with a relu activation function
- A convolutional layer with a relu activation function
- A maxpooling layer

These blocks were stacked until a desired accuracy of about 70% was obtained. This accuracy was obtained after stacking 3 VGG blocks. A 4th VGG block was added but did not seem to improve the performance as much.

The optimizer used was Adam optimizer. Dropout and Image augmentation were used to develop the final model.

2 Lessons Learnt

- Although the assignment clearly stated that changes should be made incrementally, this lesson was learnt the hard way. Since huge changes were introduced at some point leading to the deterioration of accuracy. A new model needed to be created from scratch since the offending change was not easily recognized.
- In order to obtain a descent accuracy, it involves a lot of experimentation with the architecture

3 Final Model

Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 32, 32, 32)	896
conv2d_1 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18496
conv2d_3 (Conv2D)	(None, 16, 16, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_4 (Conv2D)	(None, 8, 8, 64)	36928
conv2d_5 (Conv2D)	(None, 8, 8, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 64)	0
conv2d_6 (Conv2D)	(None, 4, 4, 64)	36928
conv2d_7 (Conv2D)	(None, 4, 4, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 64)	0
dropout (Dropout)	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 10)	1290
=====		
Total params: 247,466		
Trainable params: 247,466		
Non-trainable params: 0		
=====		