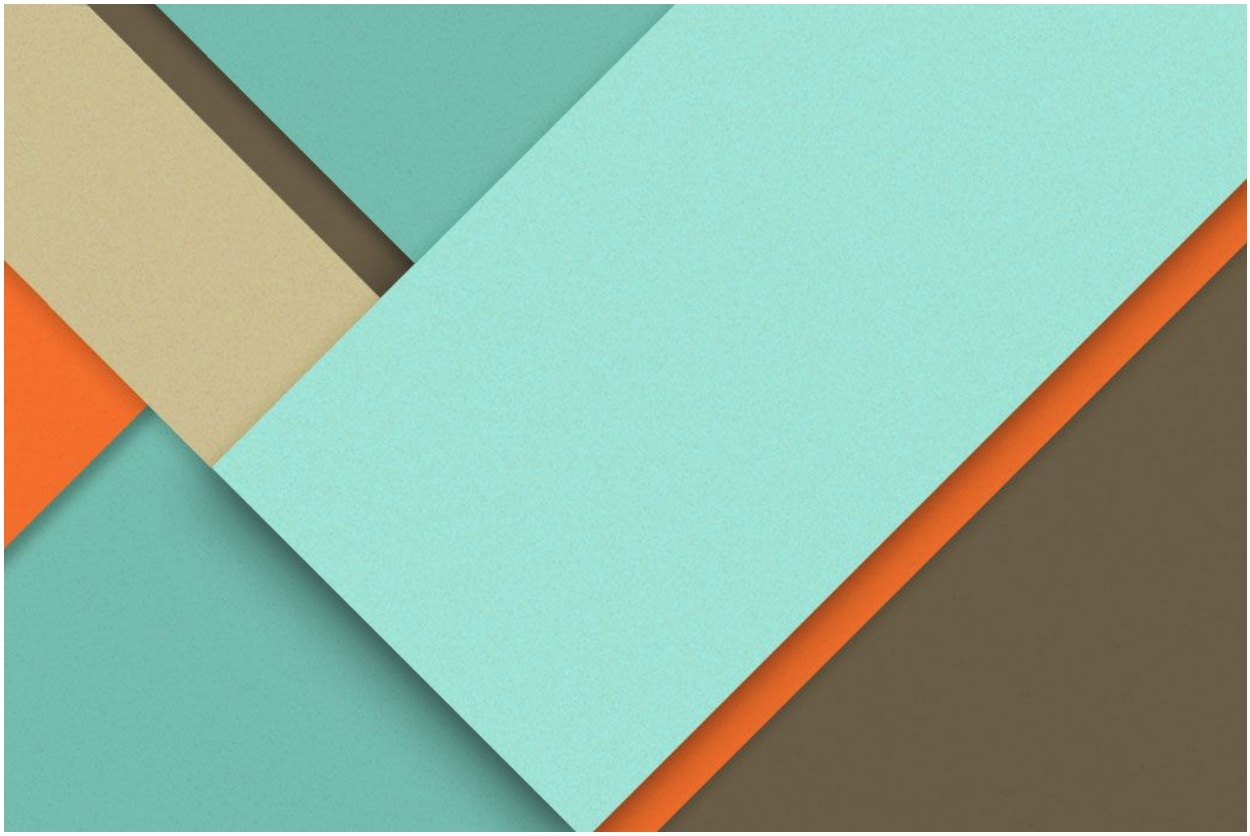


Split Kernels

21.10.2019

Nikhil Sai Bukka (IMT2016091)

Vaishnavi kalva (IMT2016078)

Arunaav (IMT2016108)

Viraj Bharadwaj (IMT2016093)

Overview:

The main objective of our project is to implement a CNN (with and without Separable kernels) from scratch to classify CIFAR-100 dataset.

Motivation:

In general size of the CNN model and computations done by the model are so large that they can't be incorporated into a smaller hardware systems. So to reduce the size and no of computations by the CNN model split kernels are used. Here we are trying to achieve the same results with and without using the split kernels.

Dataset:

The dataset we used is **CIFAR-100** and the description of the dataset is as follows:

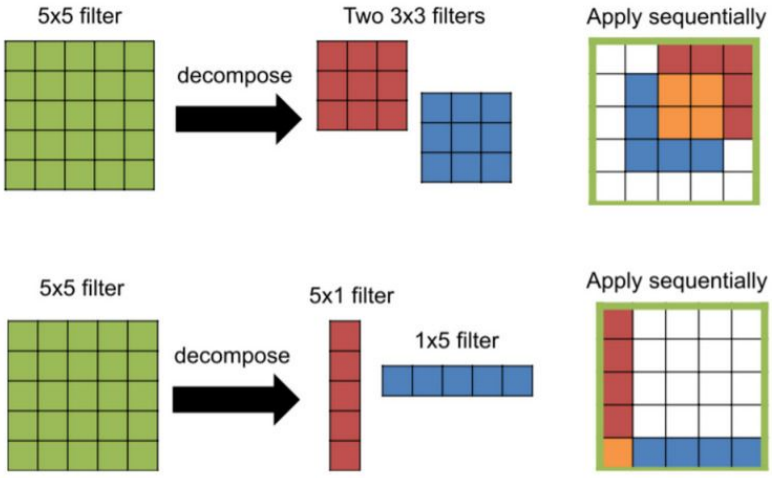
- **No of Classes: 100**
- **No of Images per class: 600**
- There are **500 training images** and **100 testing images** per class.
- Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs).
- **Sample classes**: aquarium fish, orchids,bottle etc..

Separable Kernels

A spatial separable convolution simply divides a kernel into two, smaller kernels. The most common case would be to divide a 3x3 kernel into a 3x1 and 1x3 kernel, like so:

$$\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

Now, instead of doing one convolution with 9 multiplications, we do two convolutions with 3 multiplications each (6 in total) to achieve the same effect. With less multiplications, computational complexity goes down, and the network is able to run faster.



Splits kernels helps in decreasing no of parameters to calculate.But the issue with separable kernels is that all the kernels cannot be separated into two parts.

Part -A:

For the first part of the project we used a 13 layered Deep Convolutional neural network.

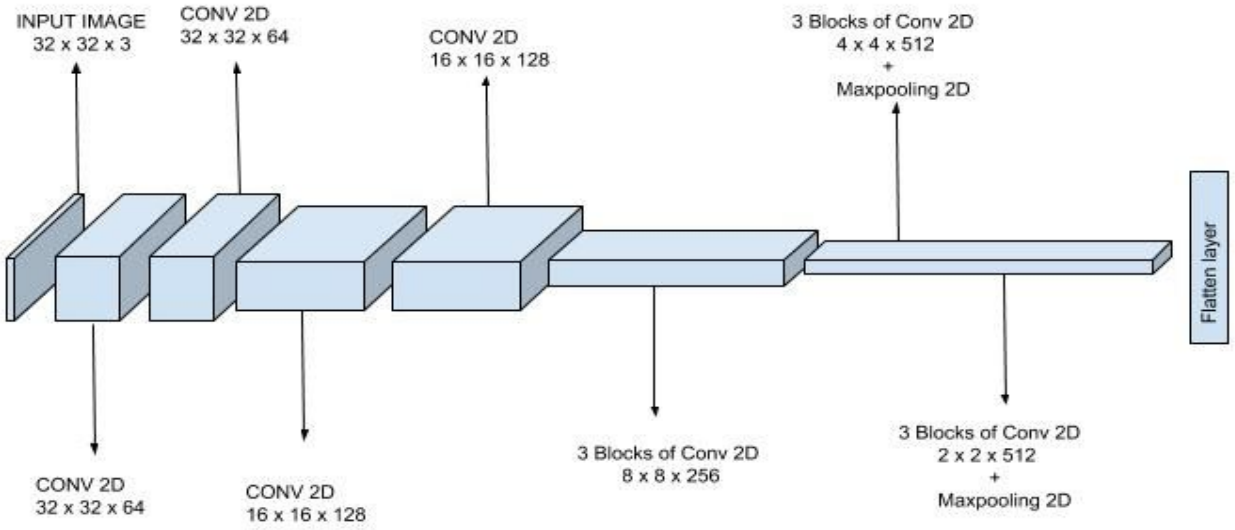
Training Procedure:

- Test and training images are converted into 32 bit float values. Both the train and test images are normalized.
- We used a Data generator to generate more images by rotating, shifting the training data images.
- Default Hyperparameters:
    - batch_size = 128
    - max _epoches = 250
    - learning_rate = 0.1
    - lr_decay = 1e-6
    - lr_drop = 20
    - momentum=0.9
- We used categorical_crossentropy as our loss function. Which is defined as

$$H_p(y) := -\sum_i p_i log(y_i)$$

y_i is the predicted probability value for class i and p_i is the true probability of that class.

Architecture:



An Image of size 32 x 32 x3 is given as input to model. As the depth of the model is increased the no of filters are increased by a factor of 2 and size of the feature map is decreased by a factor of 2 . After the Flatten layer there is a dense layer which classifies the images into one of the 100 classes.

Part -B:

The Architecture is same as the part A but instead of one normal (3 x 3) kernels we used two (3 x 1) and (1 x 3) kernels and trained the model again.

Results:

Time comparison between two architectures :

Average time taken by our architecture for each epoch without using split kernels is 90 seconds and with split kernels is 78 seconds.

Accuracy comparison :

State of art model has an accuracy of 91.3%. The maximum accuracy we could attain without split kernel is 69.16% and by using split kernels we observed a drop in accuracy and the maximum accuracy we got is 47.15% .We trained both of them using same architecture and the same parameters.

References:

- https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728
- https://en.wikipedia.org/wiki/Separable_filter
- https://benchmarks.ai/cifar-100