MobileNet

By 2016, researchers had developed compact CNNs like SqueezeNet that were much smaller than VGGNet or GoogleNet. However, these models had two key limitations:

(A) Model size ≠ computation efficiency

SqueezeNet had fewer parameters but still required a lot of FLOPS (computations) which was not ideal for real-time inference on mobile or embeded devices.

(B) No control over speed-vs- accuracy trade-off.

There was no simple way to tune how fast or how accurate you wanted the model to be depending on hardware constraints.

So, Google researchers proposed a paper titled "MobileNets: Efficient Convolutional Neural Network for Mobile Vision Applications" in year 2017.

* The core innovation was Depthwise Separable Convolutions

Instead of using normal convolutional layers (which are computationally expensive), MobileNet introduced Depthwise Separable Convolutions, breaking a standard convolution to two separate steps:

- Step 1 : Depthwise Convolution (Filtering)
  - ⇒ This step applies a single convolution filter to each input channel independently. If we have 64-channel input, this step applies 64 separate filters.
  - ⇒ It filters the spatial information (like finding edges or textures) within each channel but it doesnot combine information accross different channels.
- Step2 : The Pointwise Convolution (Combining)
  - ⇒ This is a simple 1×1 convolution that operates on the output of depthwise step. It takes the 64 separate channel output and "mixes" them together to create new features. It is 100% focused on combining channel information thus capturing cross-channel information.

Scanned with OKEN Scanner

Why is this so much more efficient?

Imagine we have a 128 X 128 X 64 input feature map and we want to produce a 128 X 128 X 128 output using a 3X3 kernel.

⇒ Standard Convolution

→ We need 128 different 3X3 filters.

→ Total calculations (FLOPs) are roughly: (3X3 X 64 X 128) X (128 X 128) ≈ 1·2 Billion flops.

⇒ Depthwise Separable Convolution

→ Step 1 (Depthwise): We need 64 different 3X3X1 filters.

→ Step 2 (Pointwise): We need 128 different 1X1X64 filters.

Total calculations are the sum of both steps:

→ Depthwise: (3X3X64) X (128 X 128) ≈ 9·4 million

→ Pointwise: (1X1X64X128) X (128X128) ≈ 134 million

* Total ≈ 143 Million FLOPs.

By factoring the convolution, the computational cost got reduced by ~8-9 times.

The model used in the ImageNet challenge had total 4·2M parameters and ~569M flops. Its model size was ~17 MB. It achieved top-1 accuracy ~70-71·1· (comparable to VGG16 but much smaller/faster).

⇒ In the coming years, MobileNet paved a way for MobileNetV2 (2018)- introduced inverted residual blocks and linear bottlenecks, which improved accuracy and efficiency.

⇒ MobileNetV3 (2019)- added squeeze and excitation and NAS based optimization.

⇒ Influenced EfficientNet (2019)- scaling depth, width and resolution together to perform optimally.

Scanned with OKEN Scanner