VGG19 Memory and Weights

Layer	**Number of Activations (Memory)**	**Parameters (Compute)**
Input	$224 \times 224 \times 3 = 150\,K$	0
CONV3-64	$224 \times 224 \times 64 = 3.2\,M$	$(3 \times 3 \times 3) \times 64 = 1,728$
CONV3-64	$224 \times 224 \times 64 = 3.2\,M$	$(3 \times 3 \times 64) \times 64 = 36,864$
POOL2	$112 \times 112 \times 64 = 800\,K$	0
CONV3-128	$112 \times 112 \times 128 = 1.6\,M$	$(3 \times 3 \times 64) \times 128 = 73,728$
CONV3-128	$112 \times 112 \times 128 = 1.6\,M$	$(3 \times 3 \times 128) \times 128 = 147,456$
POOL2	$56 \times 56 \times 128 = 400\,K$	0
CONV3-256	$56 \times 56 \times 256 = 800\,K$	$(3 \times 3 \times 128) \times 256 = 294,912$
CONV3-256	$56 \times 56 \times 256 = 800\,K$	$(3 \times 3 \times 256) \times 256 = 589,824$
CONV3-256	$56 \times 56 \times 256 = 800\,K$	$(3 \times 3 \times 256) \times 256 = 589,824$
CONV3-256	$56 \times 56 \times 256 = 800\,K$	$(3 \times 3 \times 256) \times 256 = 589,824$
POOL2	$28 \times 28 \times 256 = 200\,K$	0
CONV3-512	$28 \times 28 \times 512 = 400\,K$	$(3 \times 3 \times 256) \times 512 = 1,179,648$
CONV3-512	$28 \times 28 \times 512 = 400\,K$	$(3 \times 3 \times 512) \times 512 = 2,359,296$
CONV3-512	$28 \times 28 \times 512 = 400\,K$	$(3 \times 3 \times 512) \times 512 = 2,359,296$
CONV3-512	$28 \times 28 \times 512 = 400\,K$	$(3 \times 3 \times 512) \times 512 = 2,359,296$
POOL2	$14 \times 14 \times 512 = 100\,K$	0
CONV3-512	$14 \times 14 \times 512 = 100\,K$	$(3 \times 3 \times 512) \times 512 = 2,359,296$
CONV3-512	$14 \times 14 \times 512 = 100\,K$	$(3 \times 3 \times 512) \times 512 = 2,359,296$
CONV3-512	$14 \times 14 \times 512 = 100\,K$	$(3 \times 3 \times 512) \times 512 = 2,359,296$
CONV3-512	$14 \times 14 \times 512 = 100\,K$	$(3 \times 3 \times 512) \times 512 = 2,359,296$
POOL2	$7 \times 7 \times 512 = 25\,K$	0
FC	4096	$4096 \times 4096 = 16,777,216$
FC	4096	$4096 \times 4096 = 16,777,216$
FC	1000	$4096 \times 1000 = 4,096,000$
Total Activations	$17.62\,M$	**Total Parameters** $= 57,668,812$

(a)

The inception module in cnns is designed to capture multi-scale information by using parallel convolutional filters of different sizes (1x1, 3x3, 5x5) along with pooling operations. These filters operate in parallel, allowing the network to learn both local and global features effectively. The outputs from these filters are concatenated, preserving the spatial dimensions while increasing the depth of the feature maps.

(b)

Naive Inception Module:

- **1x1 Convolutions**: $32 \times 32 \times 128$
- **3x3 Convolutions**: $32 \times 32 \times 192$

- **5x5 Convolutions**: $32 \times 32 \times 96$
- **3x3 Max Pooling**: $32 \times 32 \times 256$

$$\text{Total Output Size} = 32 \times 32 \times (128 + 192 + 96 + 256) = 32 \times 32 \times 672$$

Inception Module with Dimension Reduction:

- **1x1 Convolutions**: $32 \times 32 \times 128$
- **3x3 Convolutions** (with 1x1 reduction): $32 \times 32 \times 192$
- **5x5 Convolutions** (with 1x1 reduction): $32 \times 32 \times 96$
- **1x1 Convolution after Max Pooling**: $32 \times 32 \times 64$

$$\text{Total Output Size} = 32 \times 32 \times (128 + 192 + 96 + 64) = 32 \times 32 \times 480$$

(c)

Naive Inception Module:

- **1x1 Convolutions**:

$$1 \times 1 \times 256 \times 128 \times 32 \times 32 = 335,544,32$$

- **3x3 Convolutions**:

$$3 \times 3 \times 256 \times 192 \times 32 \times 32 = 1,131,524,096$$

- **5x5 Convolutions**:

$$5 \times 5 \times 256 \times 96 \times 32 \times 32 = 983,040,000$$

$$\text{Total Operations} = 2,450,136,128$$

Inception Module with Dimension Reduction:

- **1x1 Convolutions for Reduction**:

$$1 \times 1 \times 256 \times 128 \times 32 \times 32 = 335,544,32$$

- **3x3 Convolutions (after reduction)**:

$$3 \times 3 \times 128 \times 192 \times 32 \times 32 = 566,362,24$$

- **5x5 Convolutions (after reduction)**:

$$5 \times 5 \times 32 \times 96 \times 32 \times 32 = 122,880,00$$

$$\text{Total Operations} = 1,024,726,56$$

(d)

The dimension reduction version reduces the computational complexity by approximately 58.4%, making it significantly more efficient than the naive version.

$$\text{Computational Savings} = \frac{2.45 - 1.02}{2.45} \times 100 \approx 58.4\%$$