# Convolutional Neural Networks Assignment

Ahmad Salimi
AI-Med

August 22, 2020

## 1 A convolutional layer with $W \times H$ inputs, $K_1 \times K_2$ kernels, $C_{in}$ input channels, $C_{out}$ output channels, padding $P$ and stride $S$

1. Number of input elements:
$$W \times H \times C_{in}$$
2. Output width:
$$W_{out} = \lfloor \frac{W + 2P - K_1}{S} \rfloor + 1$$
Output Height:
$$H_{out} = \lfloor \frac{H + 2P - K_2}{S} \rfloor + 1$$
Number of output channels:
$$C_{out}$$
3. Number of parameters:
$$|\theta| = (K_1 \times K_2 \times C_{in} + 1) \times C_{out}$$
4. Number of multiplication:
$$K_1 \times K_2 \times W_{out} \times H_{out} \times C_{out}$$

## 2 A pooling layer with $K \times K$ kernels, stride $S_2$ and padding $P_2$

1. Number of input elements:
$$W \times H \times C_{in}$$
2. Output width:
$$W_{out} = \lfloor \frac{W + 2P_2 - K}{S_2} \rfloor + 1$$
Output Height:
$$H_{out} = \lfloor \frac{H + 2P_2 - K}{S_2} \rfloor + 1$$
Number of output channels:
$$C_{in}$$
3. Number of parameters:
$$0$$
4. Number of multiplication:
$$0$$

## 3 A fully connected layer with $N$ neurons

1. Number of input elements:
$$W \times H \times C_{in}$$
2. Output size:
$$N$$
3. Number of parameters:
$$|\theta| = (W \times H \times C_{in} + 1) \times N$$
4. Number of multiplication:
$$W \times H \times C_{in} \times N$$

## 4 Parameters placement

$W = H = 256$
$K_1 = K_2 = 3$
$P = 1$
$S = 1$
$C_{in} = 64$
$C_{out} = 128$
$K = 2$
$S_2 = 2$
$P_2 = 0$
$N = 1000$

1. Convolutional layer:
   (a) Number of parameters:
   $$|\theta| = (K_1 \times K_2 \times C_{in} + 1) \times C_{out} = (3 \times 3 \times 64 + 1) \times 128 = 73,856$$
   (b) Number of multiplications:
   $$K_1 \times K_2 \times W_{out} \times H_{out} \times C_{out} = 3 \times 3 \times (\frac{256 + 2 - 3}{1} + 1) \times (\frac{256 + 2 - 3}{1} + 1) \times 128 = 74,908,800$$
   (c) Output size:
   $$\lfloor \frac{W + 2P - K_1}{S} \rfloor + 1 = \lfloor \frac{H + 2P - K_2}{S} \rfloor + 1 = \frac{256 + 2 - 3}{1} + 1 = 256$$
2. Pooling layer:
   (a) Number of parameters:
   $$0$$
   (b) Number of multiplications:
   $$0$$
   (c) Output size:
   $$\lfloor \frac{W + 2P_2 - K}{S_2} \rfloor + 1 = \lfloor \frac{H + 2P_2 - K}{S_2} \rfloor + 1 = \frac{256 + 0 - 2}{2} + 1 = 128$$
3. Fully connected layer:
   (a) Number of parameters:
   $$|\theta| = (W \times H \times C_{in} + 1) \times N = (256 \times 256 \times 64 + 1) \times 1000 = 4,194,305,000$$
   (b) Number of multiplications:
   $$W \times H \times C_{in} \times N = 256 \times 256 \times 64 \times 1000 = 4,194,304,000$$
   (c) Output size:
   $$N = 1000$$

# 5 Network bottleneck

```
              Layers                |      Shape      |      Size
----------------------------------------------------------------
Input: 256 x 256                    # B 1    256 256  = B * 65536
[64] Conv 3 x 3, s=1, p=1           # B 64   256 256  = B * 4194304
[64] Conv 3 x 3, s=1, p=1           # B 64   256 256  = B * 4194304
Pool 2 x 2, s=2, p=0                # B 64   128 128  = B * 1048576
[128] Conv 3 x 3, s=1, p=1          # B 128  128 128  = B * 2097152
[128] Conv 3 x 3, s=1, p=1          # B 128  128 128  = B * 2097152
Pool 2 x 2, s=2, p=0                # B 128  64  64   = B * 524288
[256] Conv 3 x 3, s=1, p=1          # B 256  64  64   = B * 1048576
[256] Conv 3 x 3, s=1, p=1          # B 256  64  64   = B * 1048576
Pool 2 x 2, s=2, p=0                # B 256  32  32   = B * 262144
[512] Conv 3 x 3, s=1, p=1          # B 512  32  32   = B * 524288
[512] Conv 3 x 3, s=1, p=1          # B 512  32  32   = B * 524288
Pool 2 x 2, s=2, p=0                # B 512  16  16   = B * 131072
[512] Conv 3 x 3, s=1, p=1          # B 512  16  16   = B * 131072
[512] Conv 3 x 3, s=1, p=1          # B 512  16  16   = B * 131072
Pool 2 x 2, s=2, p=0                # B 512  8   8    = B * 32768
Flatten                             # B 32768         = B * 32768
FC (4096)                           # B 4096          = B * 4096
FC (4096)                           # B 4096          = B * 4096
FC (2)                              # B 2             = B * 2
```

It turns out that the largest size is $B \times 4194304$. So, if we assume each number to be `float32`, the size of the largest possible batch will be as follows:

$$B = \lfloor \frac{12GB}{4194304 \times 4B} \rfloor = \lfloor \frac{12 \times 2^{30}}{2^{24}} \rfloor = 12 \times 2^6 = 768$$

# 6 Receptive Field

For each convolutional and pooling layer with kernel size $K$, stride $S$ and padding $P$, the $(i, j)$ neuron of $n^{th}$ layer, represents the following range of $n-1^{th}$ layer:

$$(Si - P : Si - P + K - 1, Sj - P : Sj - P + K - 1)$$

```
    [   i    :    i    ,    j    :    j    ]
Conv[  i-1  :    i+1  ,    j-1  :    j+1  ]
Conv[  i-2  :    i+2  ,    j-2  :    j+2  ]
Pool[ 2i-4  :   2i+5  ,   2j-4  :   2j+5  ]
Conv[ 2i-5  :   2i+6  ,   2j-5  :   2j+6  ]
Conv[ 2i-4  :   2i+7  ,   2j-4  :   2j+7  ]
Pool[ 4i-8  :   4i+15,   4j-8  :   4j+15]
Conv[ 4i-9  :   4i+16,   4j-9  :   4j+16]
Conv[ 4i-10:   4i+17,   4j-10:   4j+17]
Pool[ 8i-20:   8i+35,   8j-20:   8j+35]
Conv[ 8i-21:   8i+36,   8j-21:   8j+36]
Conv[ 8i-22:   8i+37,   8j-22:   8j+37]
Pool[16i-44:  16i+75,  16j-44:  16j+75]
Conv[16i-45:  16i+76,  16j-45:  16j+76]
Conv[16i-46:  16i+77,  16j-46:  16j+77]
```

So, the $(i, j)$ neuron of last convolutional layer, represents the following range of the input image:

$$(16i - 44 : 16i + 75, 16j - 44 : 16j + 75)$$