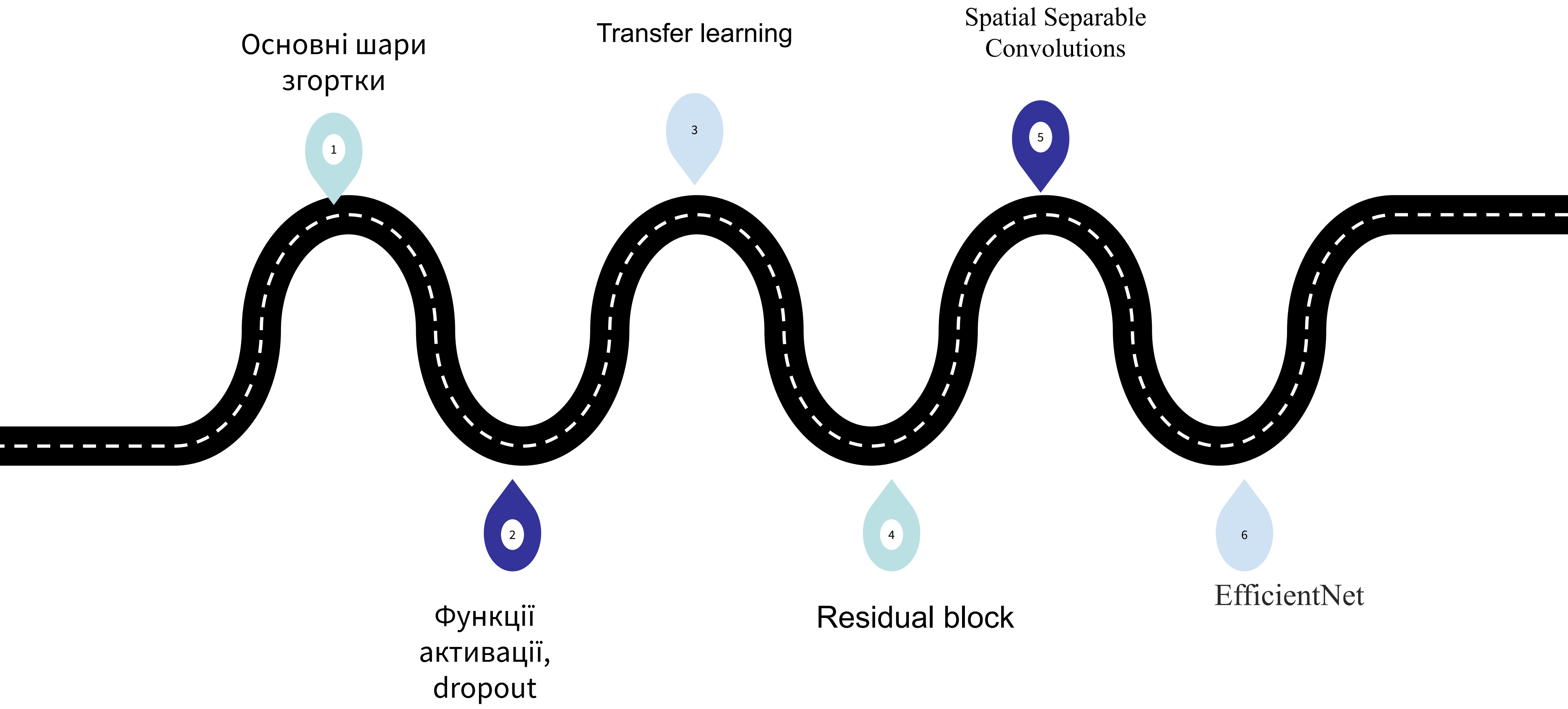
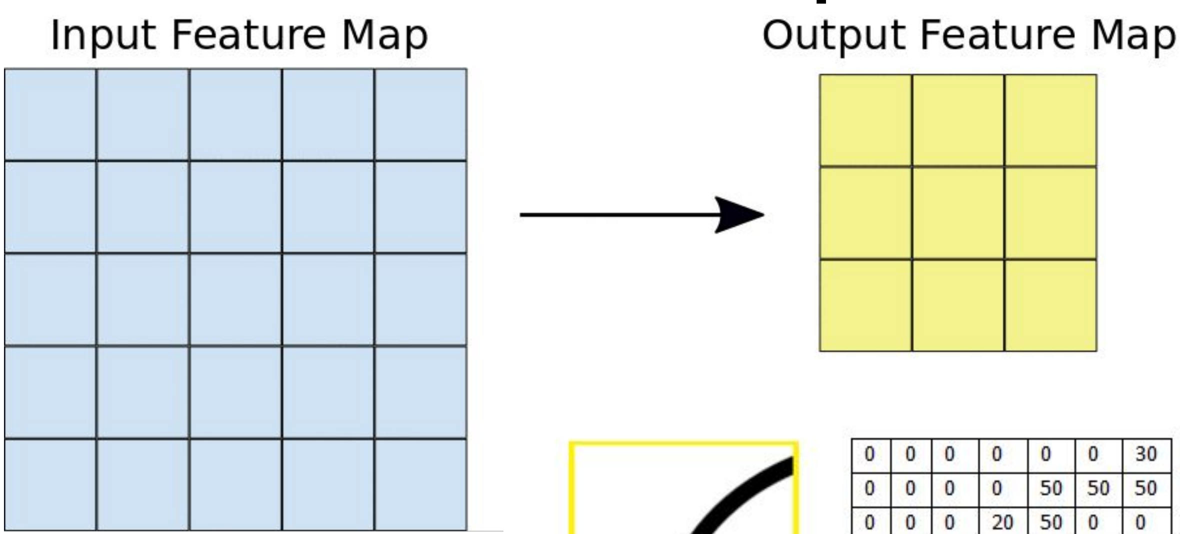
Розпізнавання образів.

Згорткові нейронні мережі

Сьогодні на лекції



Згортка



Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

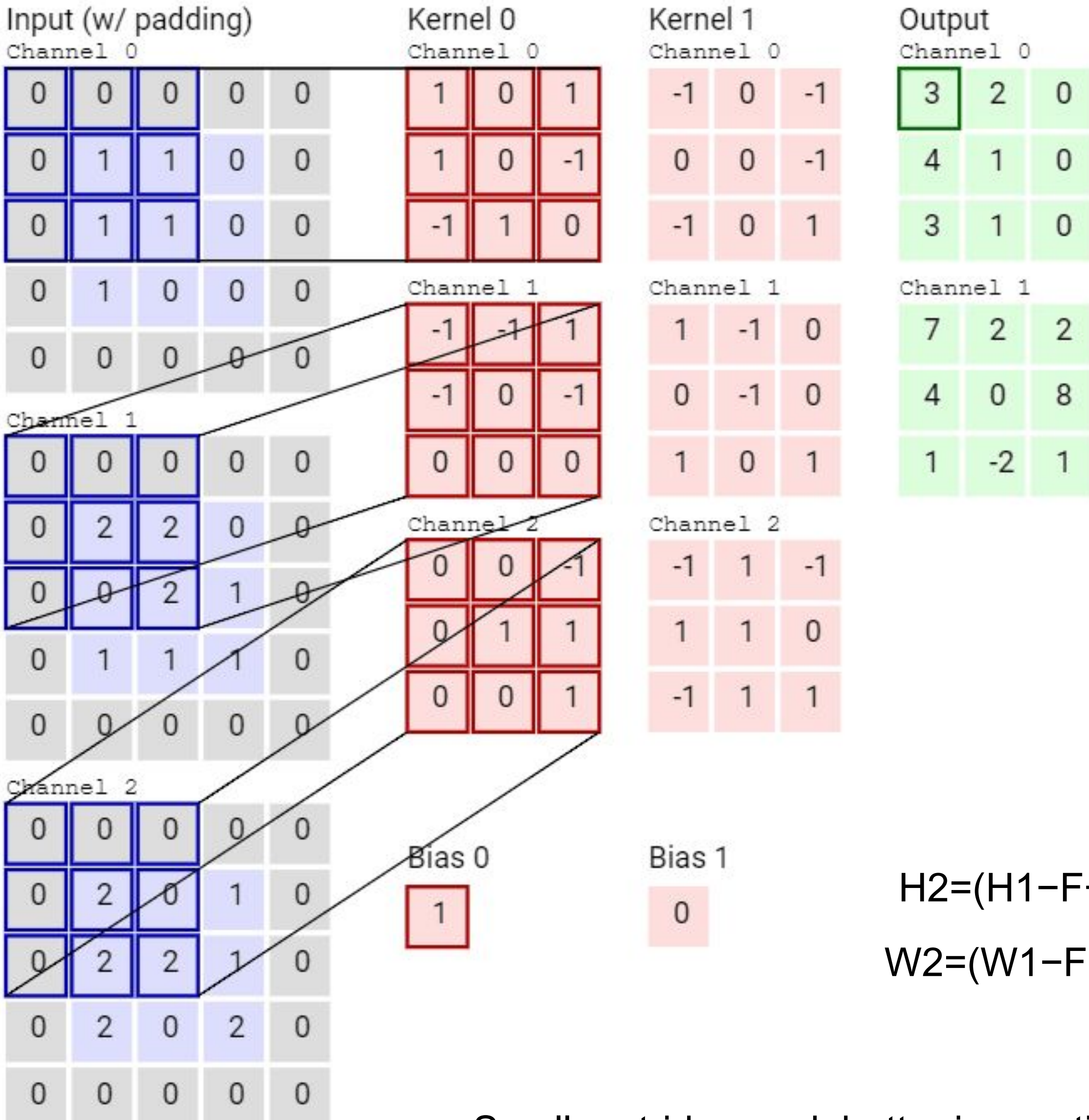
Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)



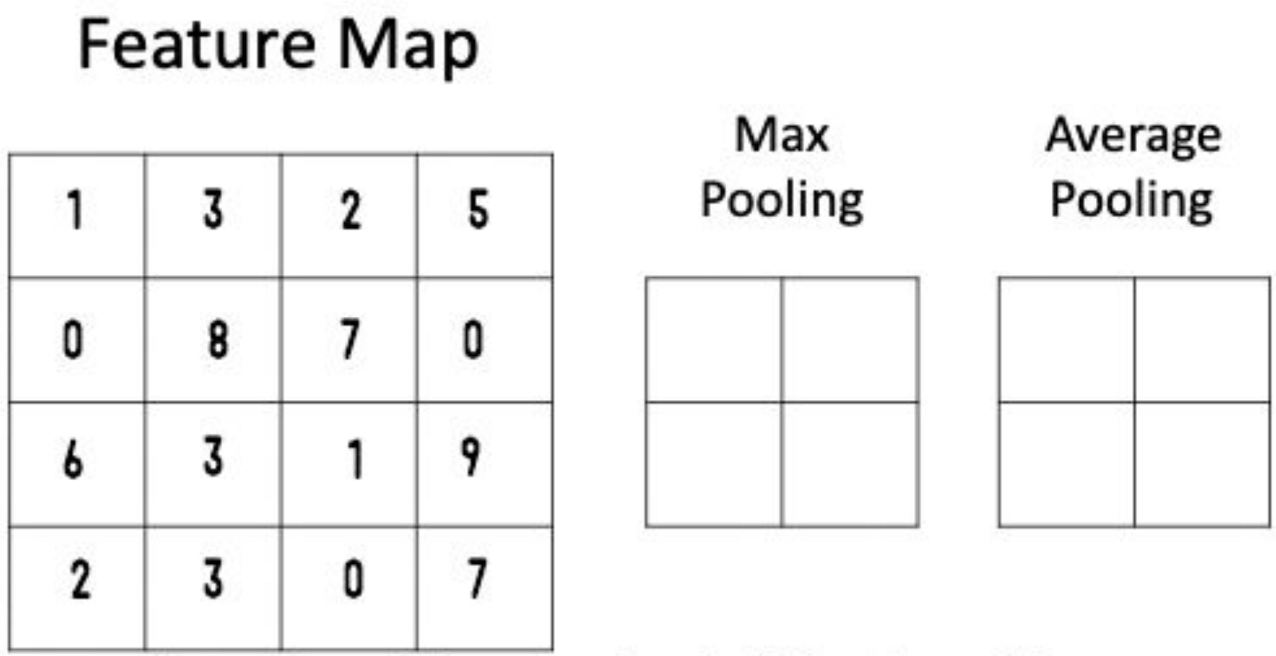
feature map

$$H2=(H1-F+2P)/S+1$$

$$W2=(W1-F+2P)/S+1$$

Smaller strides work better in practice.

Пулінг (підвибірка)



Max pooling and Average pooling of a 4x4 input map with a 2x2 pooling window and stride = 2
thedatabus.io

S=2

При застосуванні згортки 5x5x3, 256 разів, вихідна карта ознак матиме скільки каналів?

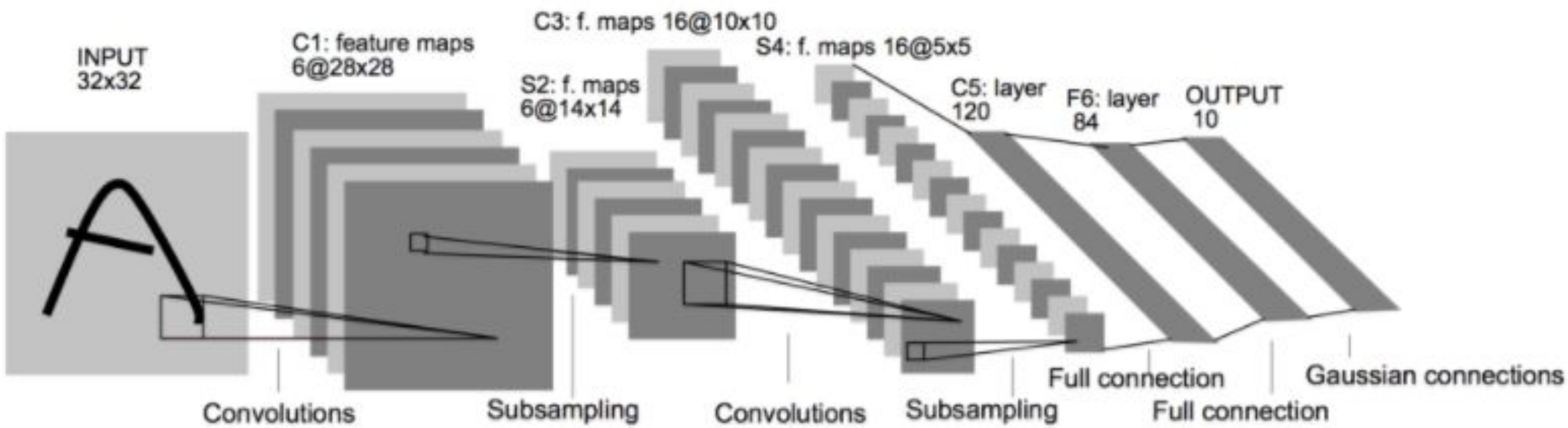
3

5

256

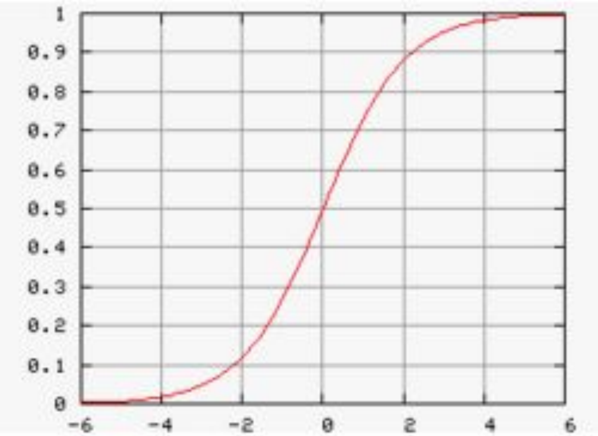
8

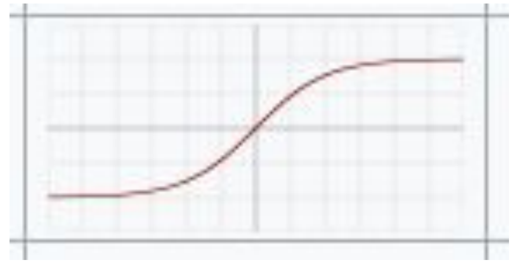
LeNet5



INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

Функції активації



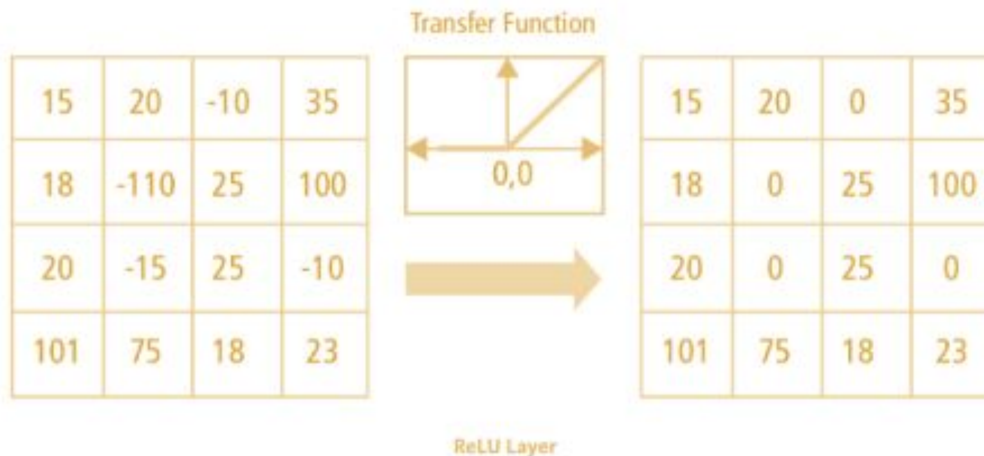


$$\begin{cases} 0 & \text{if } x \le 0 \\ x & \text{if } x > 0 \end{cases}$$

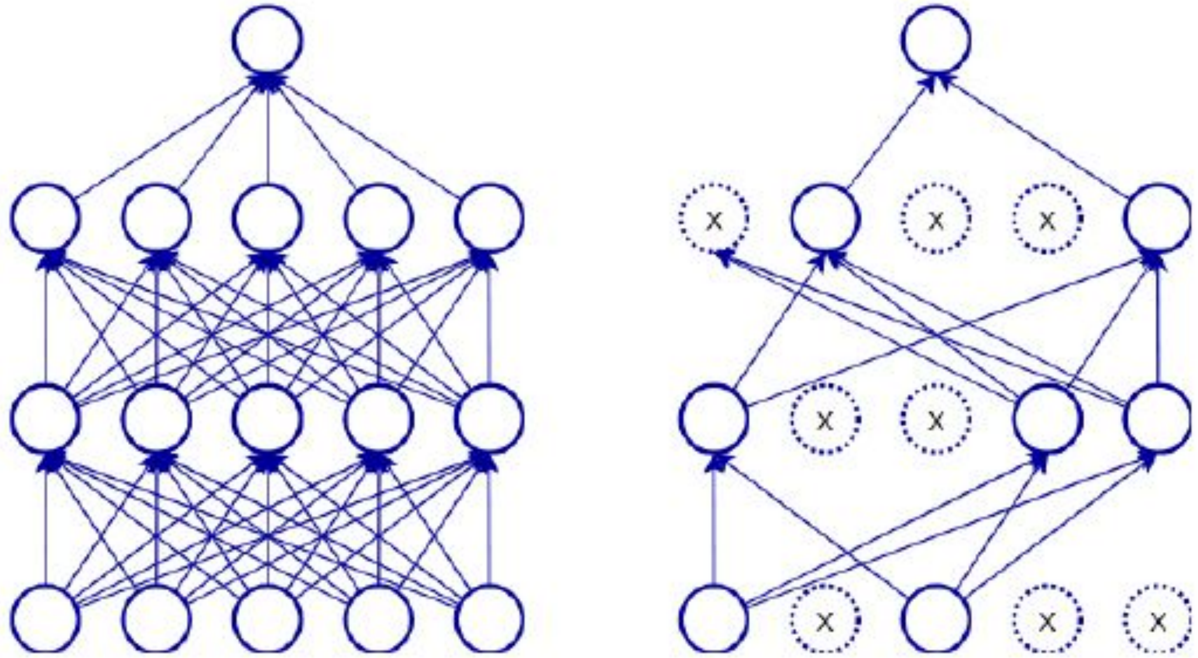
$$= \max\{0, x\}$$

$$\begin{cases} \alpha x & \text{if } x < 0 \\ x & \text{if } x \ge 0 \end{cases}$$

with parameter α



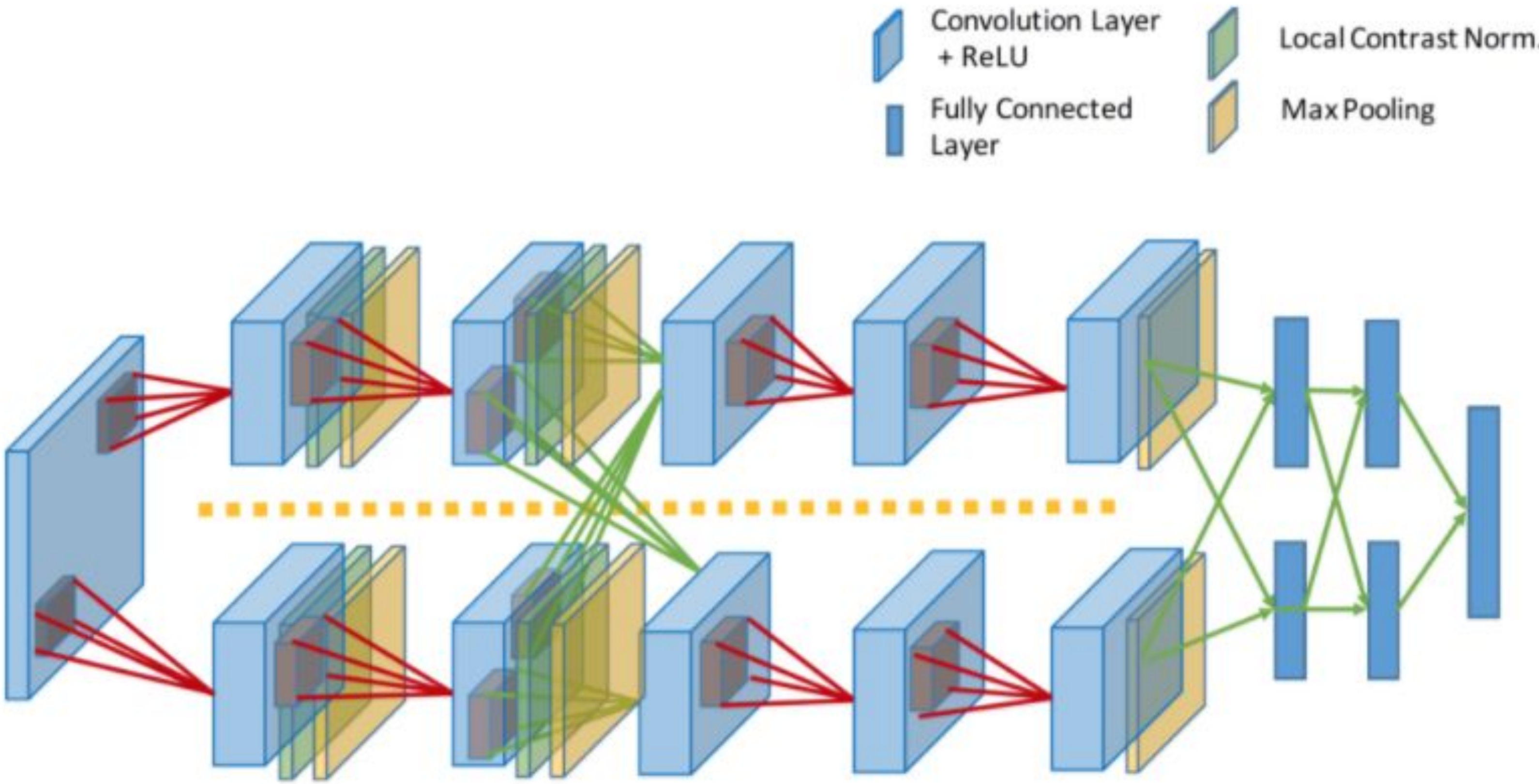
Dropout



$$\hat{w_{ij}} = \begin{cases} w_{ij}, & \text{with } P(c) \\ 0, & \text{otherwise} \end{cases} \qquad \hat{\mathbf{w}}_j = \begin{cases} \mathbf{w}_j, & \text{with } P(c) \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

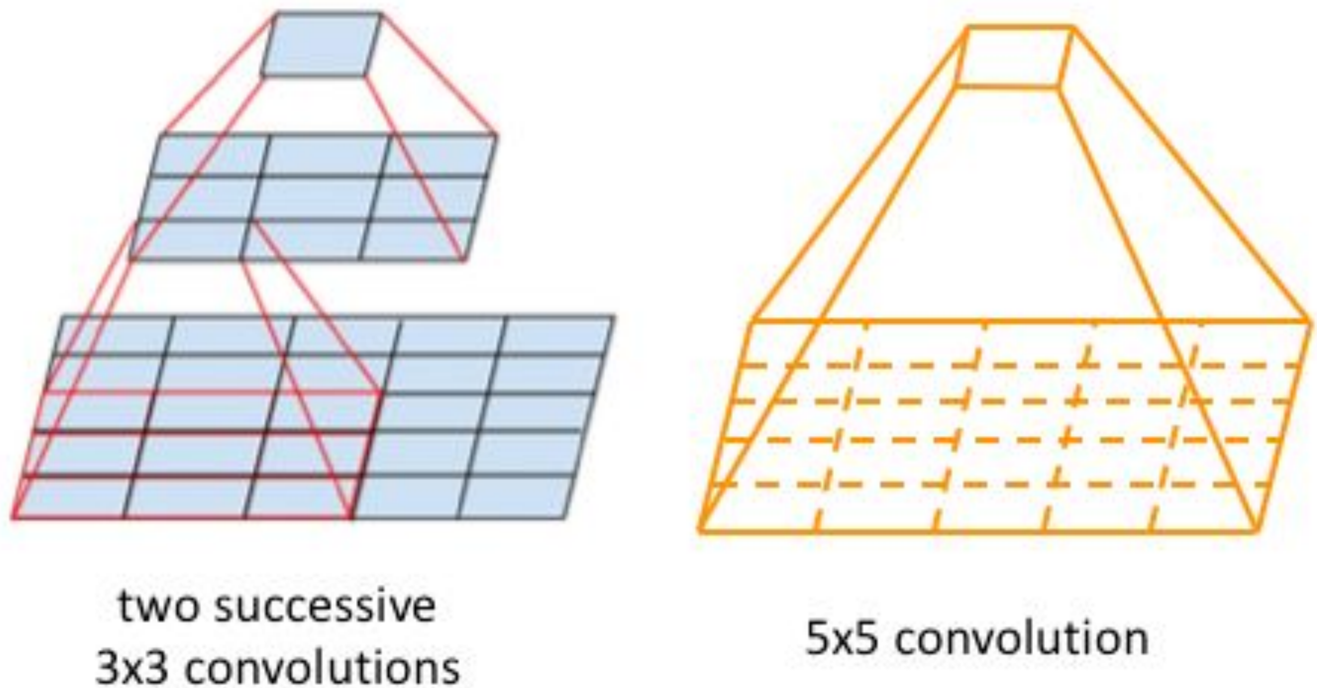
1,0 до 0,1 з кроком 0,1.

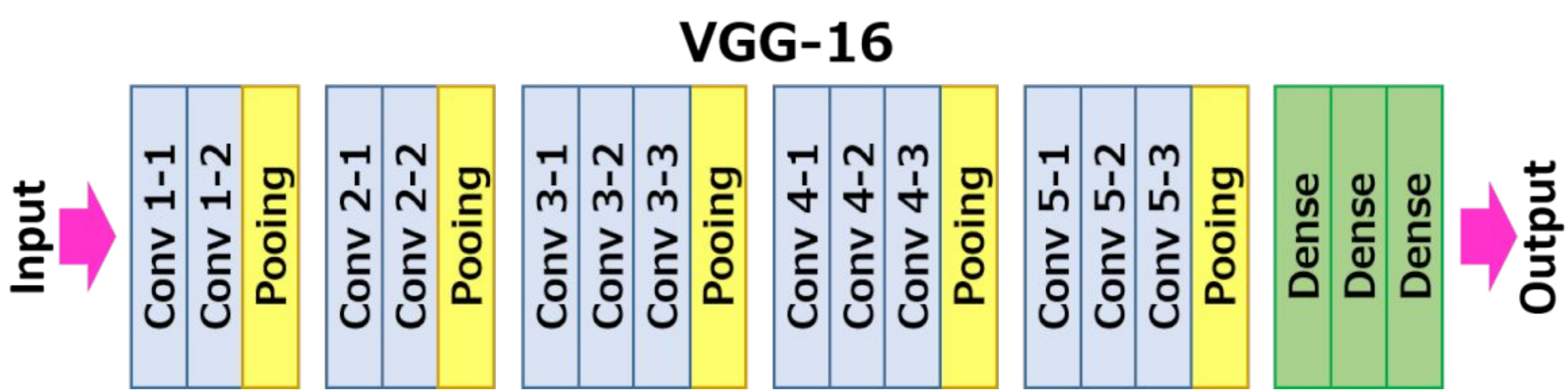
AlexNet


Convolution Layer + ReLU
Local Contrast Norm.
Fully Connected Layer
Max Pooling

10 6

VGGnet


two successive
3x3 convolutions
5x5 convolution


VGG-16
Input
Conv 1-1
Conv 1-2
Pooing
Conv 2-1
Conv 2-2
Pooing
Conv 3-1
Conv 3-2
Conv 3-3
Pooing
Conv 4-1
Conv 4-2
Conv 4-3
Pooing
Conv 5-1
Conv 5-2
Conv 5-3
Pooing
Dense
Dense
Dense
Output

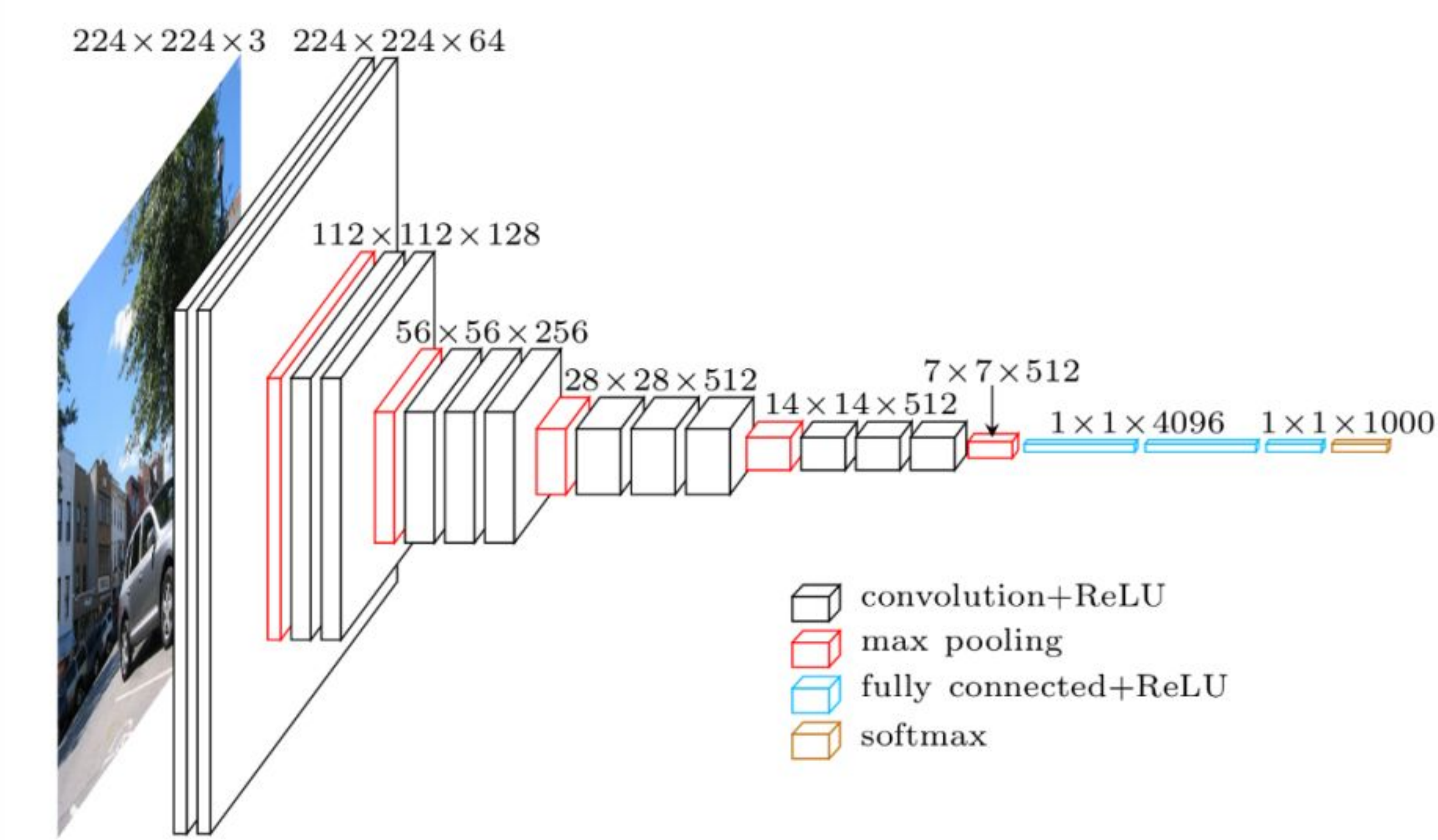
Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

- Z –вектор
- Zi – елементи ветора
- К – кількість класів

$$\begin{bmatrix} P(\text{cat}) \\ P(\text{dog}) \end{bmatrix} = \sigma(\begin{bmatrix} 1.2 \\ 0.3 \end{bmatrix}) = \begin{bmatrix} \frac{e^{1.2}}{e^{1.2}+e^{0.3}} \\ \frac{e^{0.3}}{e^{1.2}+e^{0.3}} \end{bmatrix} = \begin{bmatrix} 0.71 \\ 0.29 \end{bmatrix}$$

VGGnet16

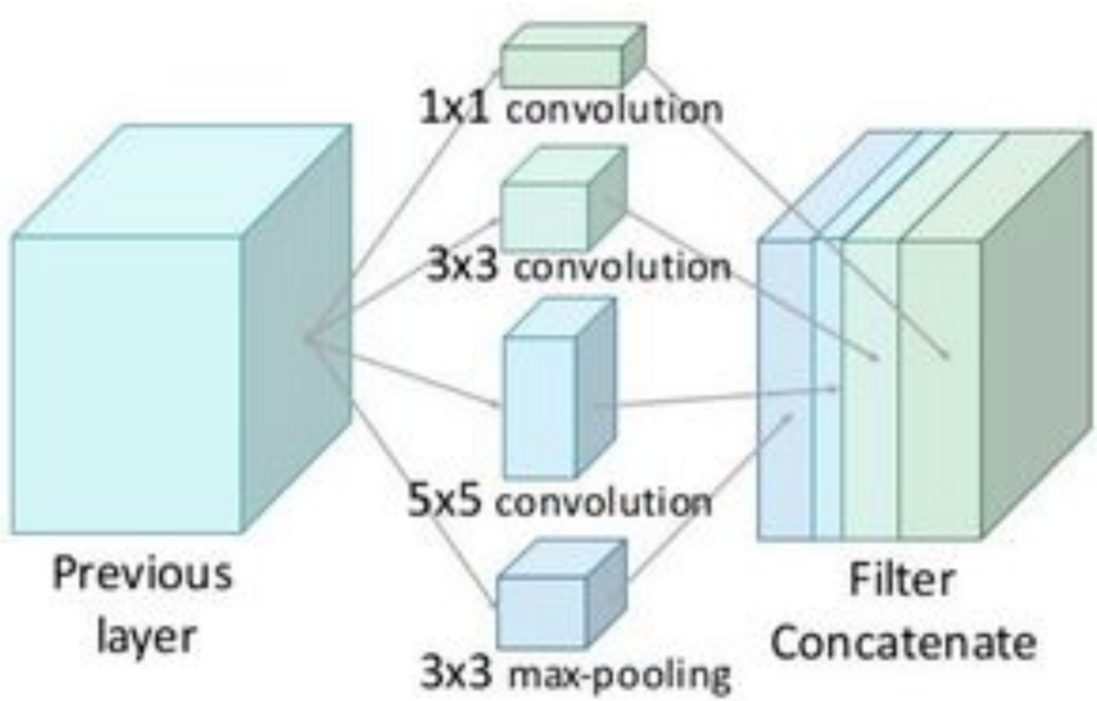


Дуже повільна швидкість навчання.
Сама архітектура мережі важить надто багато

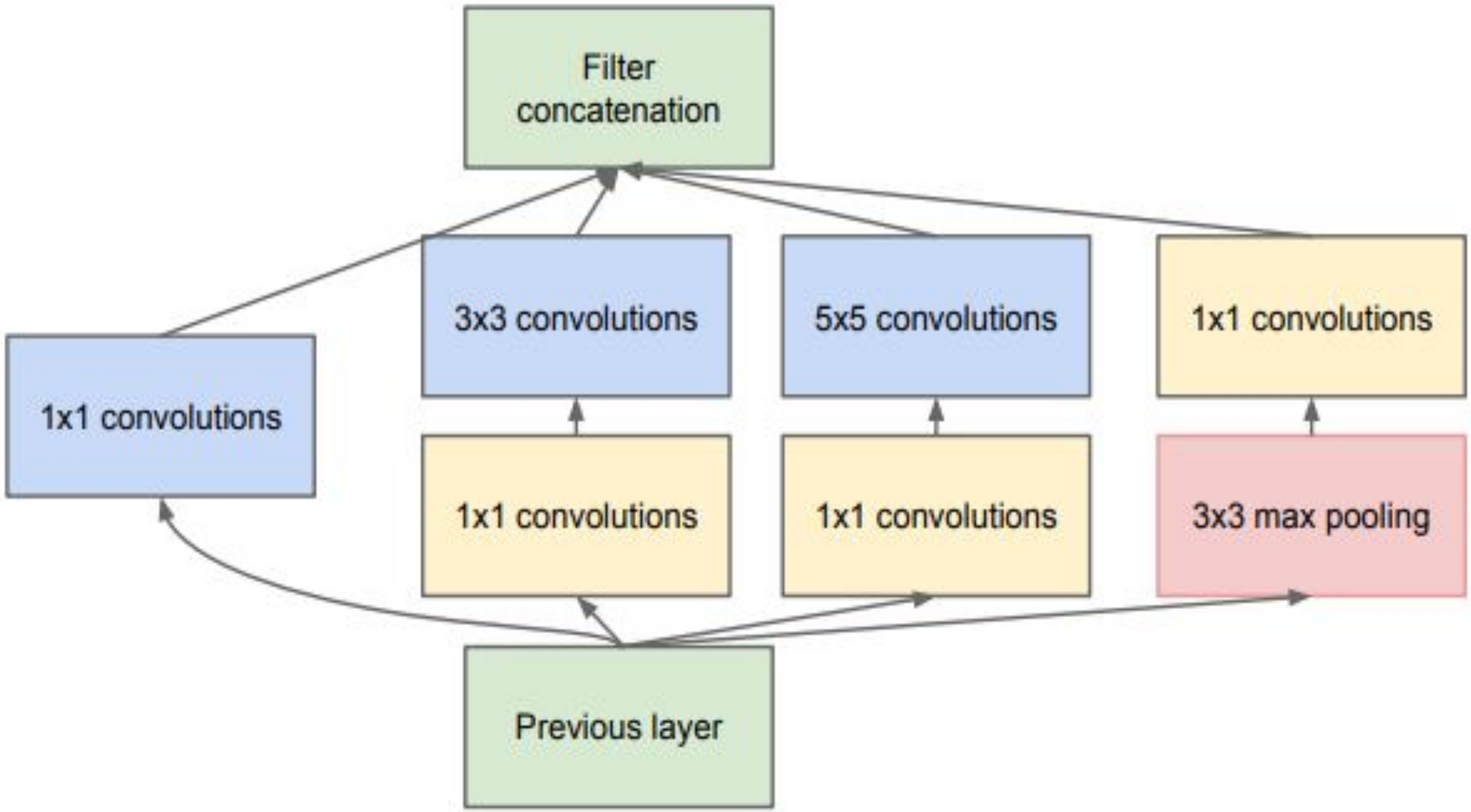
Transfer learning

augmentation

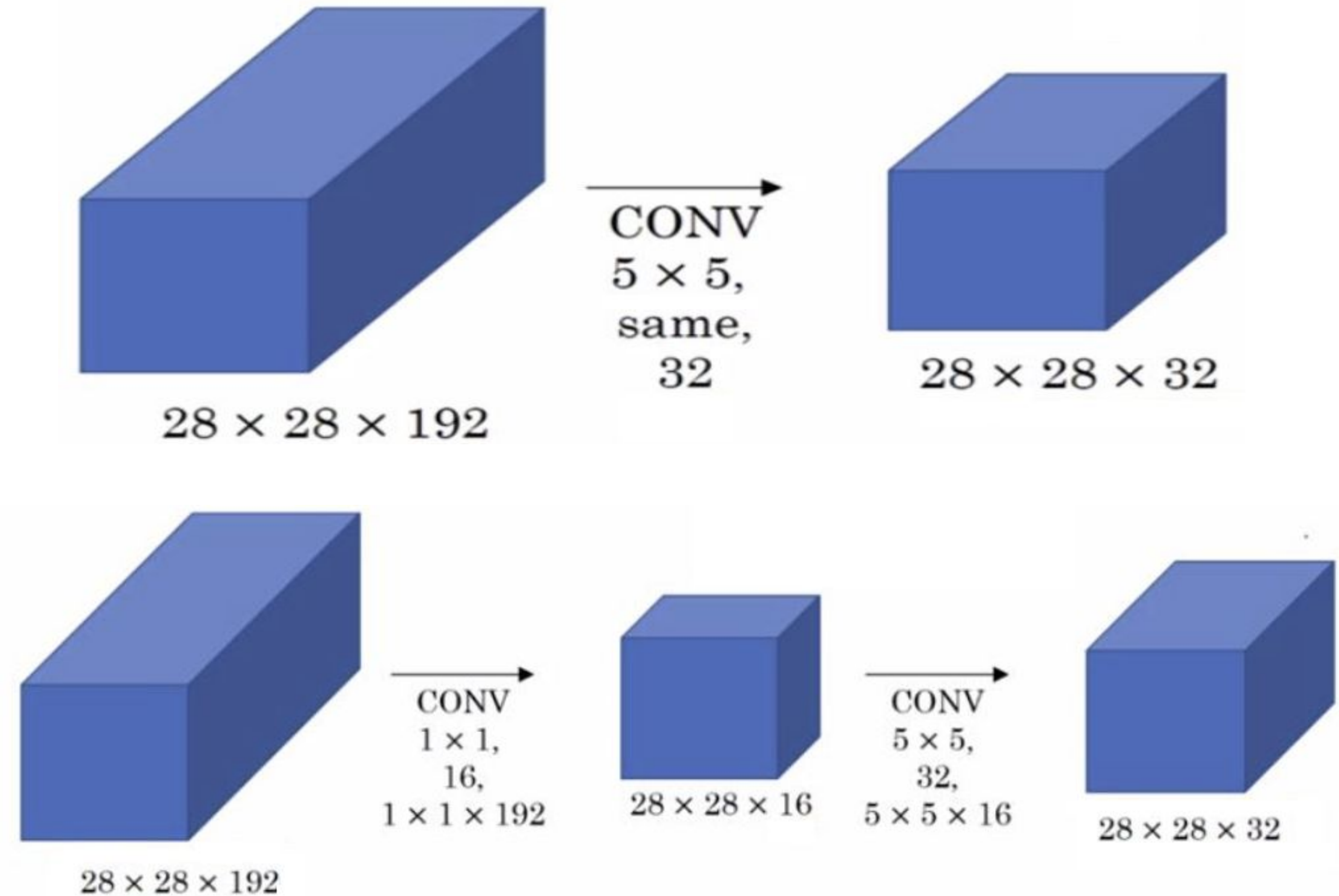
Inception-v1 (GoogLeNet)



Inception модуль

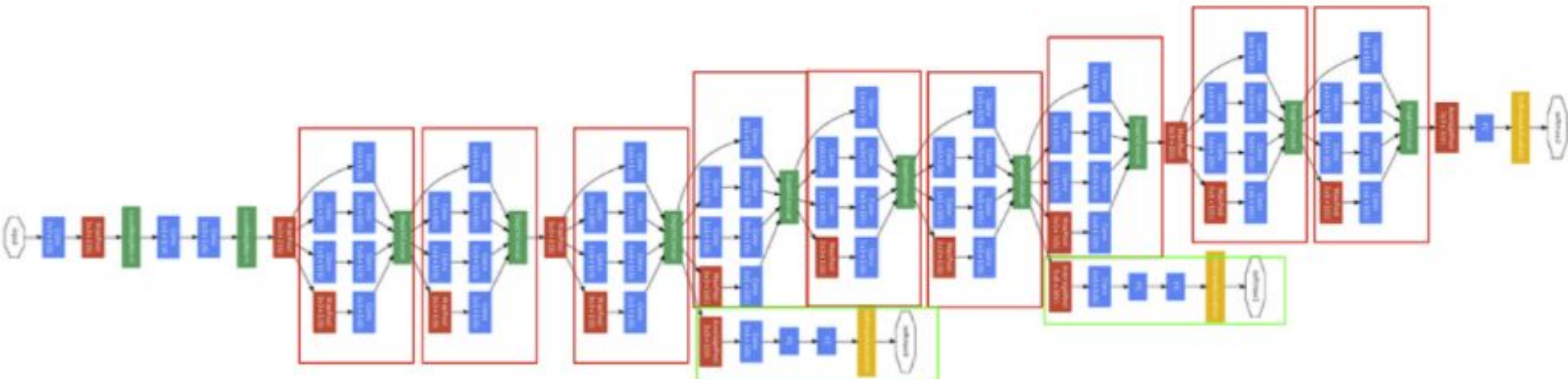


Згортка 1x1



CONV
5 × 5,
same,
32
28 × 28 × 192
28 × 28 × 32
CONV
1 × 1,
16,
1 × 1 × 192
28 × 28 × 16
CONV
5 × 5,
32,
5 × 5 × 16
28 × 28 × 192
28 × 28 × 32

Inception-v1



SGD

$$L_i = \frac{1}{2}(y_i - wx_i)^2$$

$$g_{SGD} = \frac{\partial L}{\partial w} = -(y_i - wx_i)x_i$$

$$w = w - lr \cdot g_{SGD}$$

miniBatch (розмір 4)

$$L = \frac{1}{2}\left((y_1 - wx_1)^2 + \cdots + (y_4 - wx_4)^2\right)$$

$$L = \frac{1}{2}\left((y_5 - wx_5)^2 + \cdots + (y_8 - wx_8)^2\right)$$

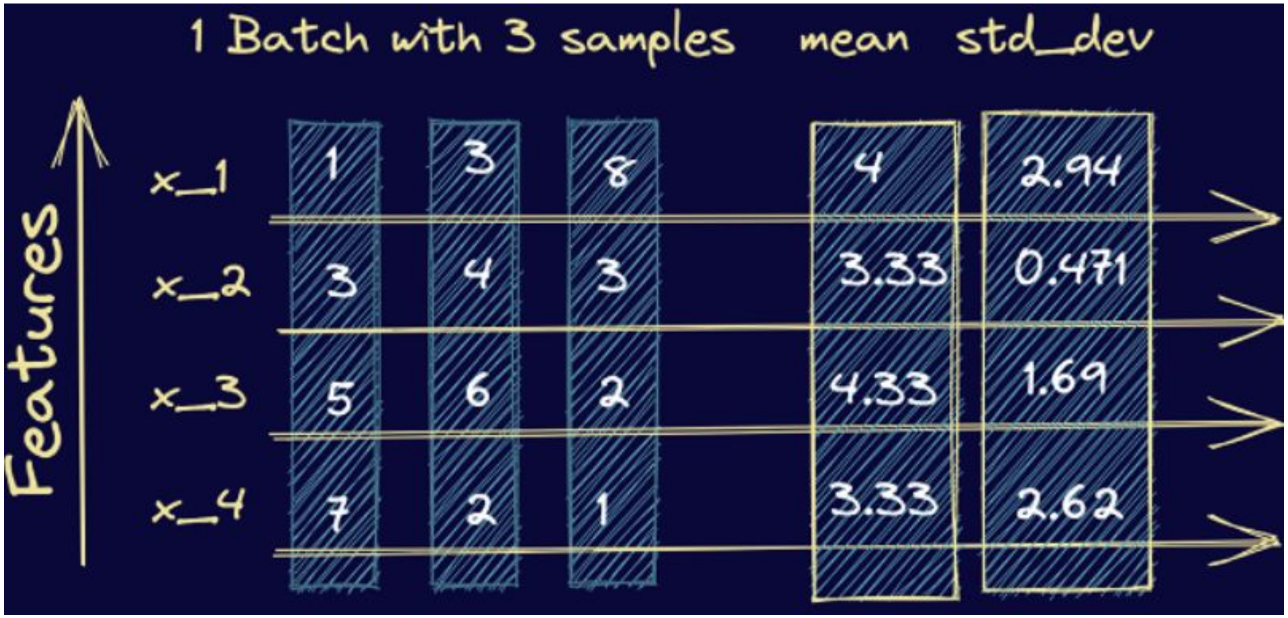
.....

$$g_{mini-batch} = \frac{\partial L}{\partial w} = -\left((y_1 - wx_1)x_1 + \cdots + (y_4 - wx_4)x_4\right)$$

$$w = w - lr * g_{mini-batch}$$

Batch Normalization

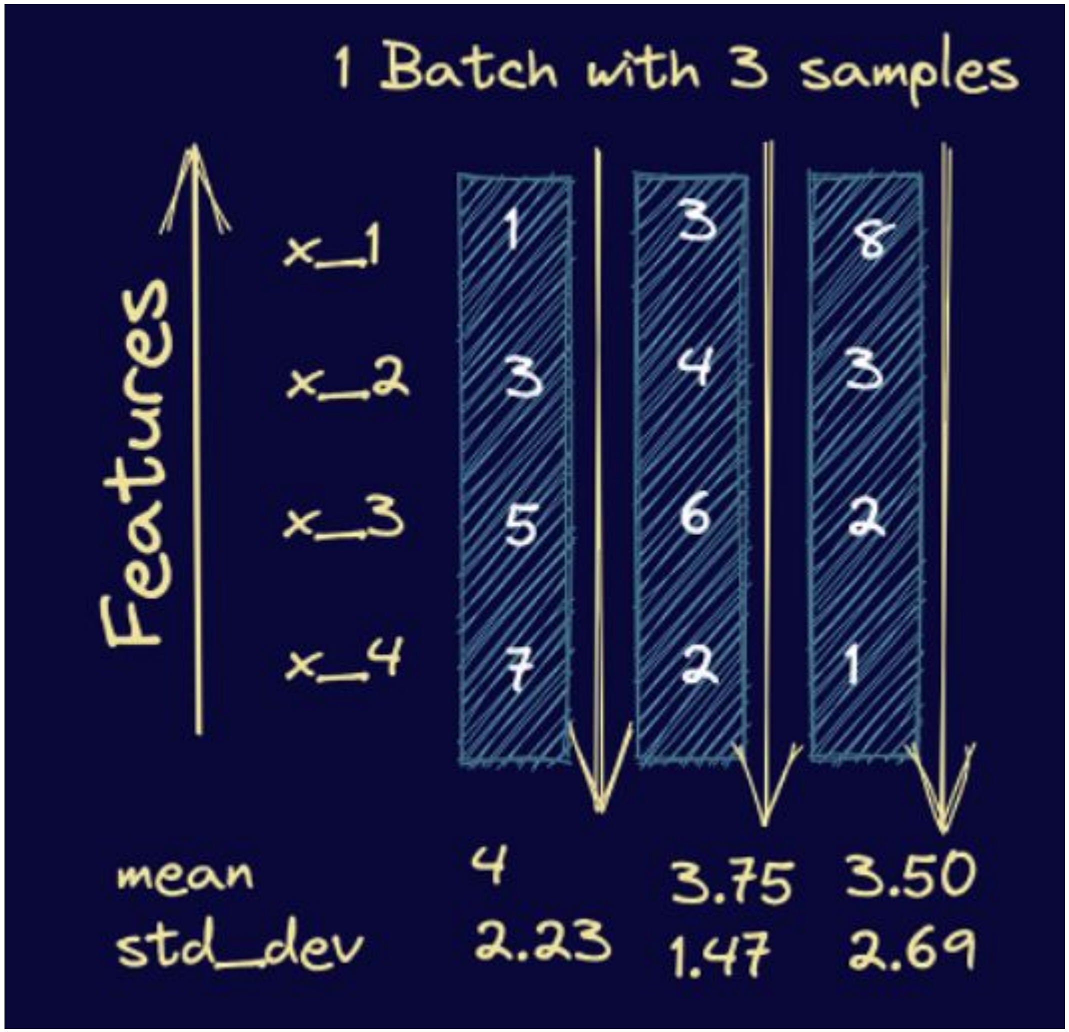
$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \text{, and } \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \qquad x = (x^{(1)}, \ldots, x^{(d)})$$

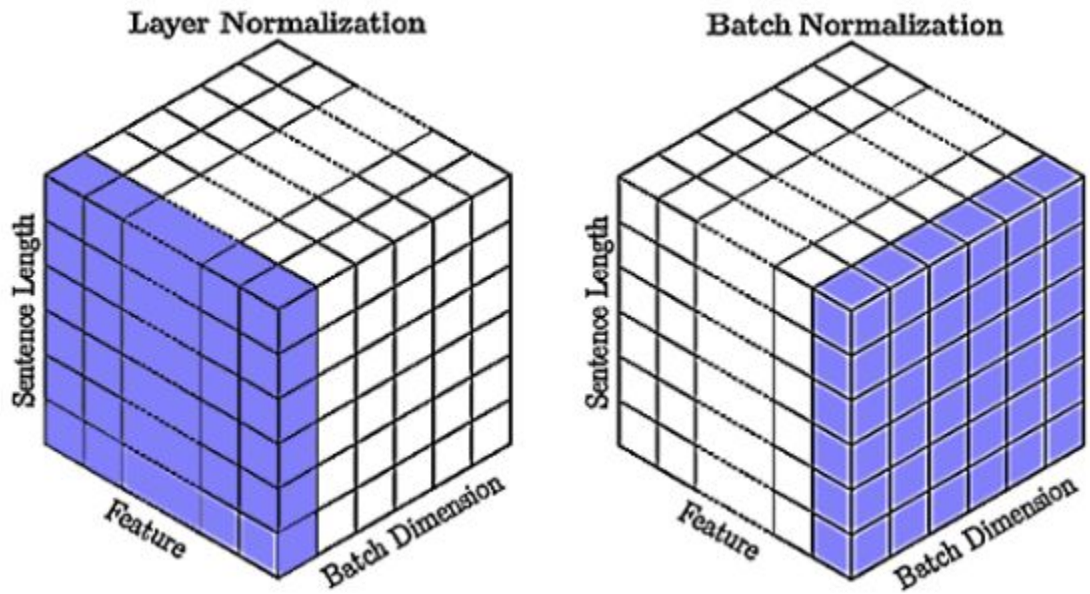


$$\hat{x}_i^{(k)} = \frac{x_i^{(k)} - \mu_B^{(k)}}{\sqrt{\left(\sigma_B^{(k)}\right)^2 + \epsilon}} \text{, where } k \in [1, d] \text{ and } i \in [1, m]$$

$$y_i^{(k)} = \gamma^{(k)} \hat{x}_i^{(k)} + \beta^{(k)}$$

Layer Normalization

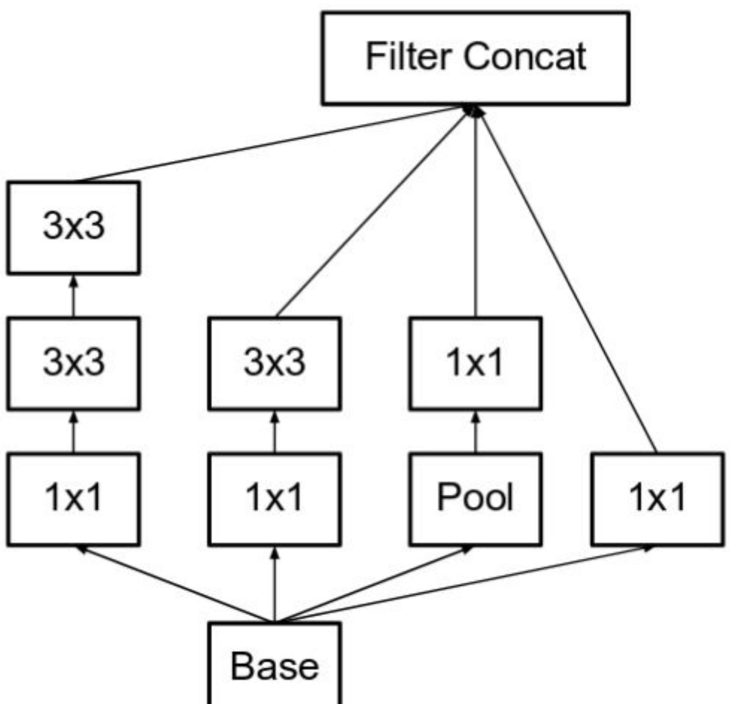


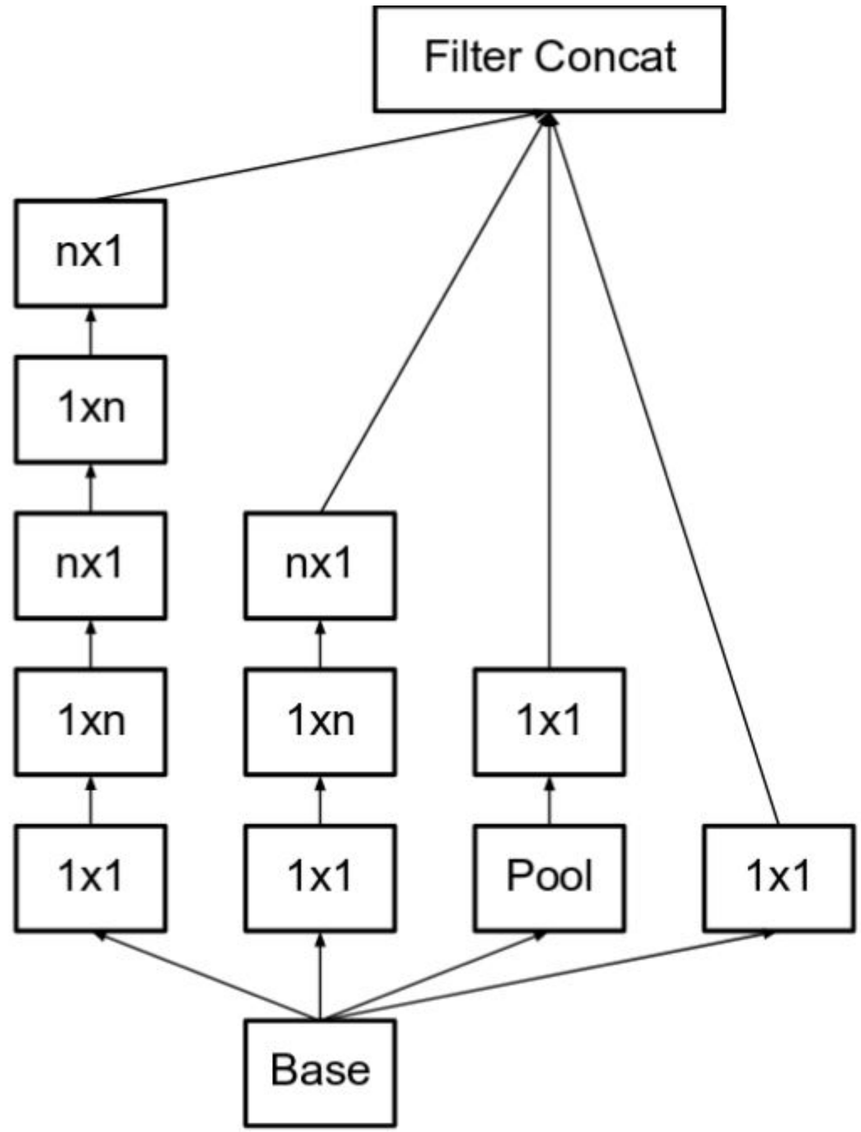

Layer Normalization
Sentence Length
Feature
Batch Dimension
Batch Normalization
Sentence Length
Feature
Batch Dimension

Inception-v2 і Inception-v3

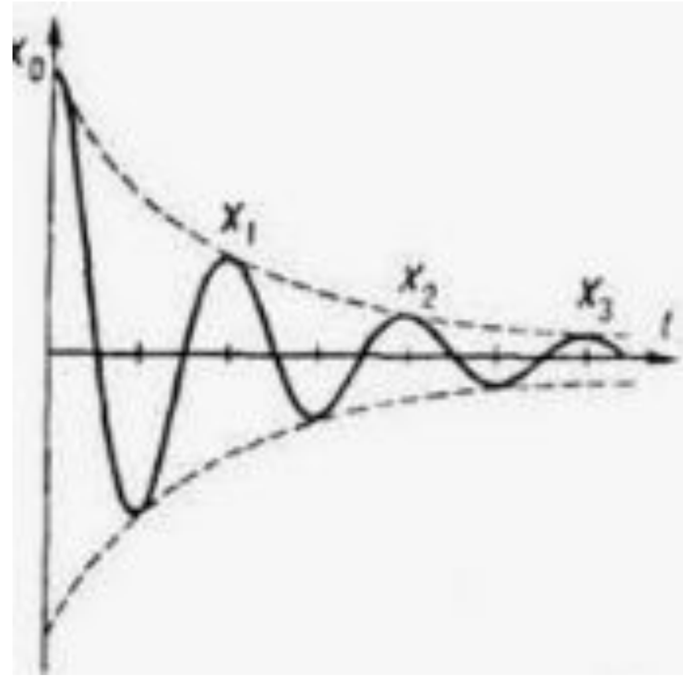
Inception-v2
- декомпозиція шару з фільтром 5x5 на два шари 3x3.
- Batch Normalization

Inception-v3
- декомпозиція фільтрів NxN двома послідовними фільтрами 1xN і Nx1
- RMSProp

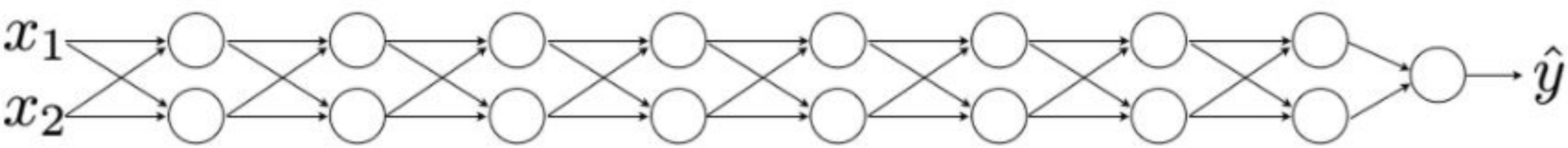




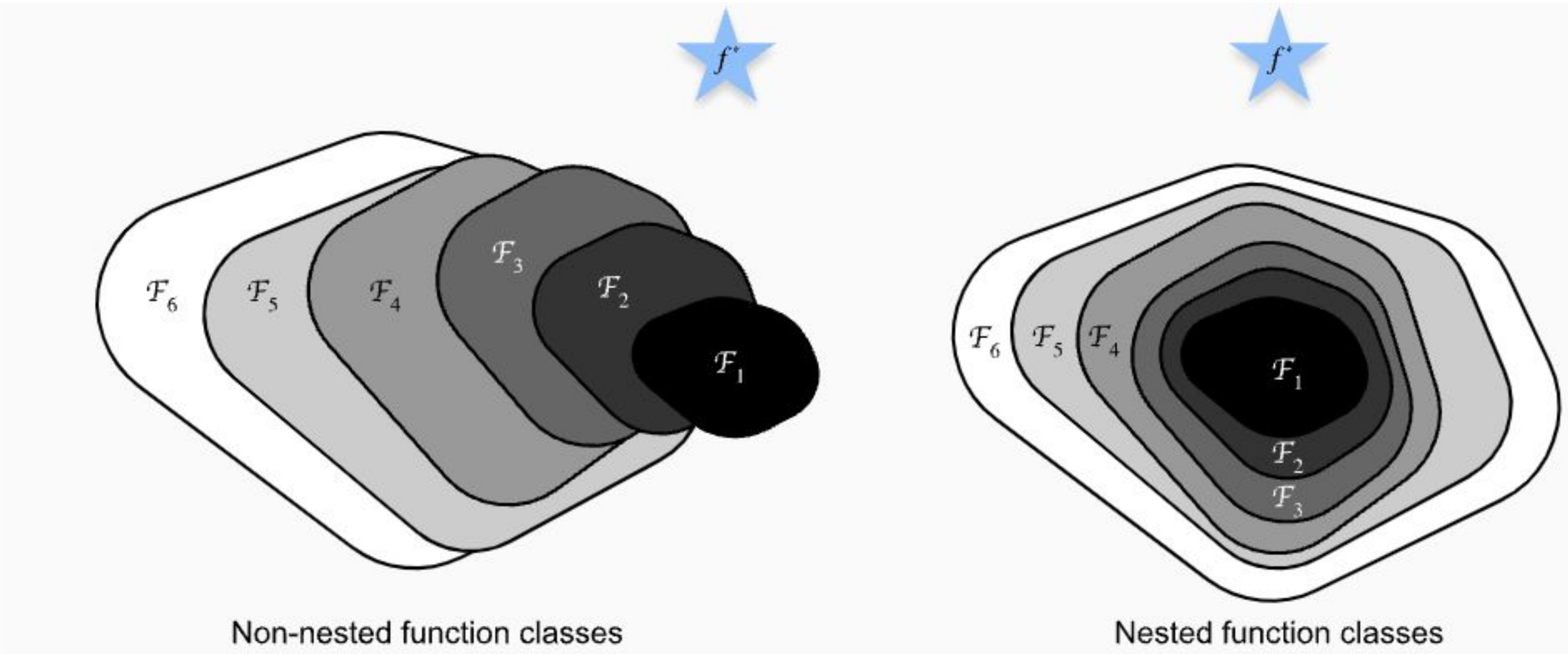
Нащо потрібна згортка 1x1?



x_0
x_1
x_2
x_3
t



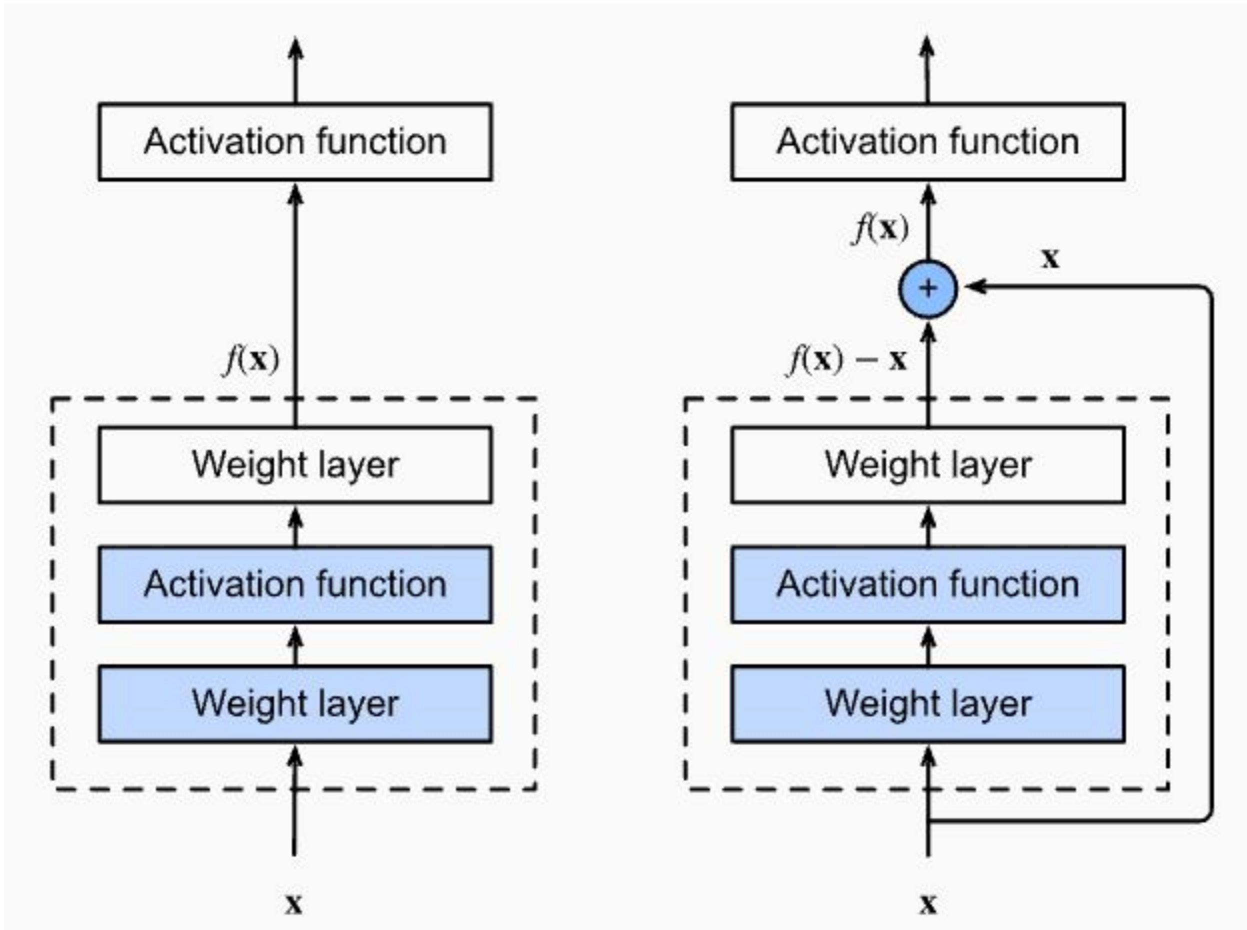
x_1
x_2
$\hat{y}$



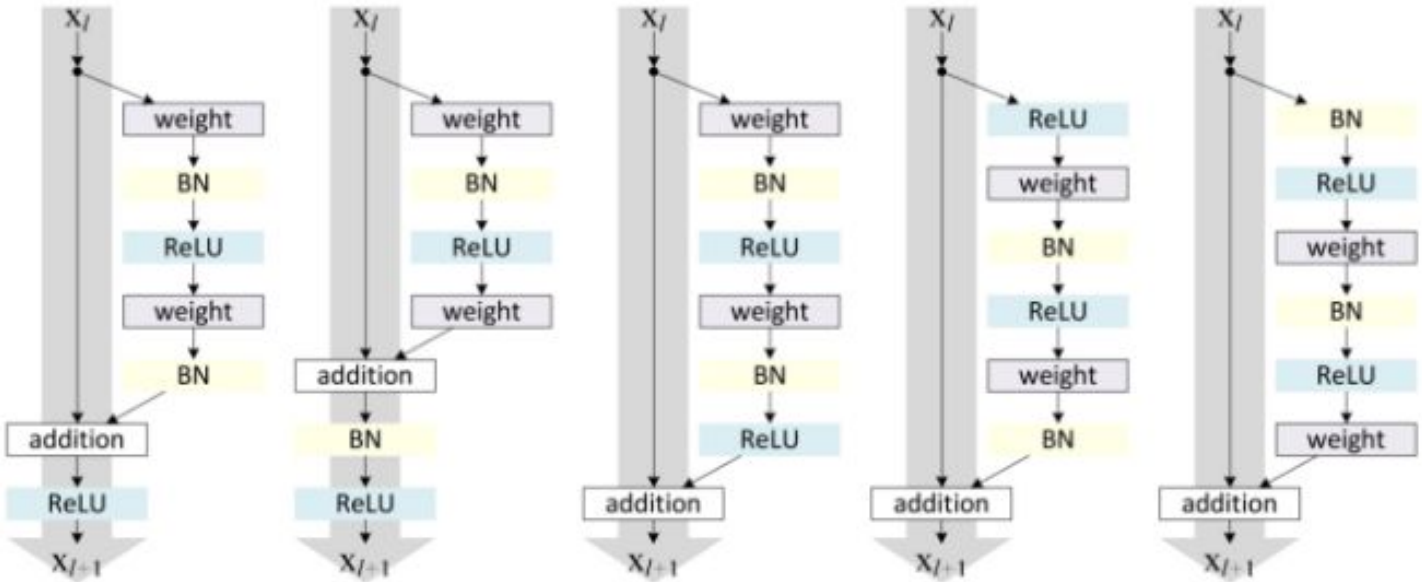
f^*
$\mathcal{F}_6$
$\mathcal{F}_5$
$\mathcal{F}_4$
$\mathcal{F}_3$
$\mathcal{F}_2$
$\mathcal{F}_1$
Non-nested function classes
f^*
$\mathcal{F}_6$
$\mathcal{F}_5$
$\mathcal{F}_4$
$\mathcal{F}_1$
$\mathcal{F}_2$
$\mathcal{F}_3$
Nested function classes

$f(x)=x$

ResNet (Residual Network)



$$H(x) = f(x) + x,$$
$$f(x) = 0$$
$$H(x) = x$$



(a) original (b) BN after addition (c) ReLU before addition (d) ReLU-only pre-activation (e) **full pre-activation**

варіанти залишкових блоків

Inception-v4 i Inception-ResNet

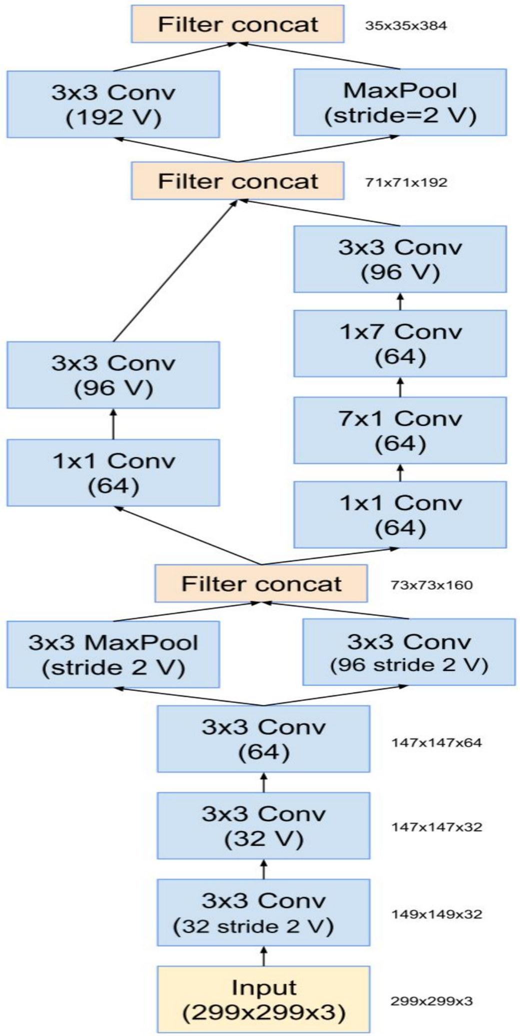


Figure 3. The schema for stem of the pure Inception-v4 and Inception-ResNet-v2 networks. This is the input part of those networks. Cf. Figures 9 and 15

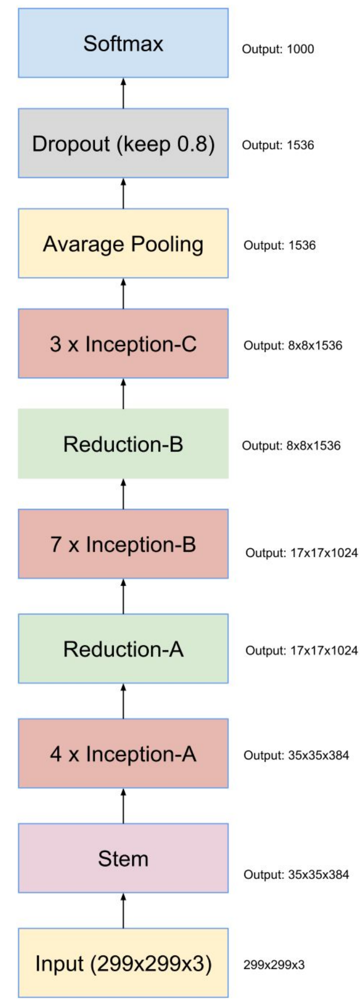
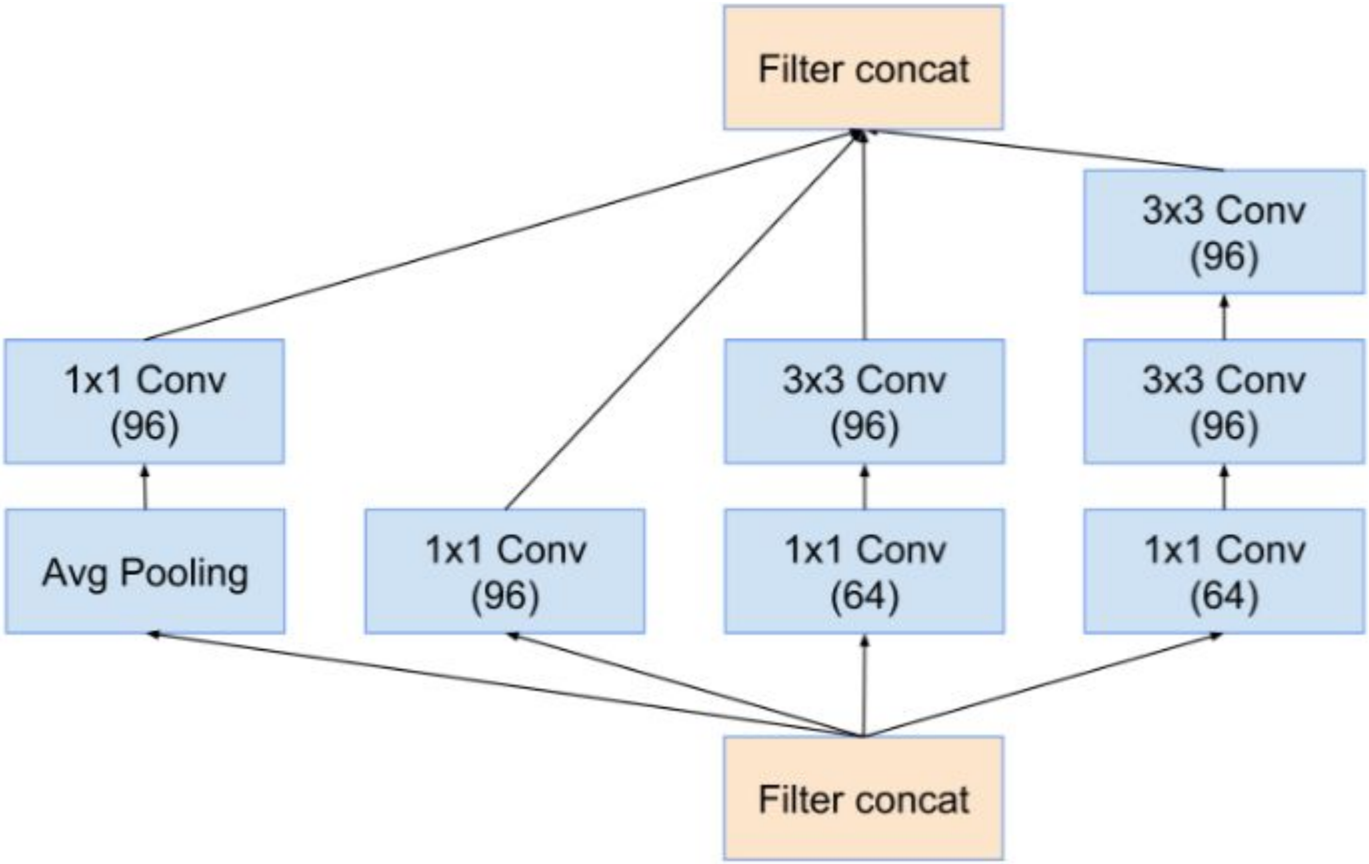


Figure 9. The overall schema of the Inception-v4 network. For the detailed modules, please refer to Figures 3, 4, 5, 6, 7 and 8 for the detailed structure of the various components.

Inception блок A



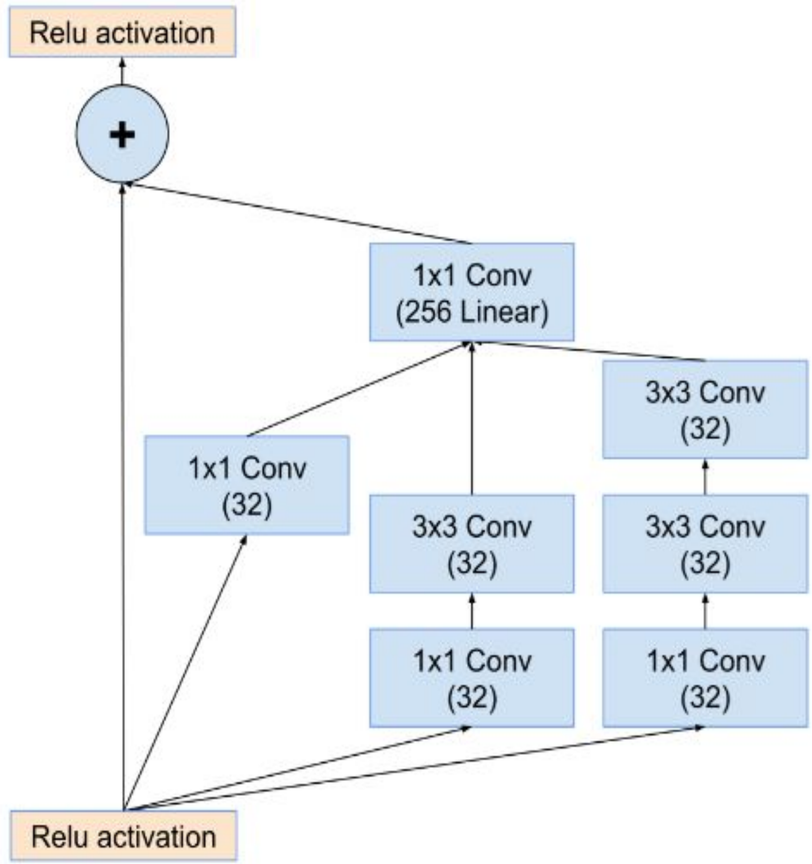
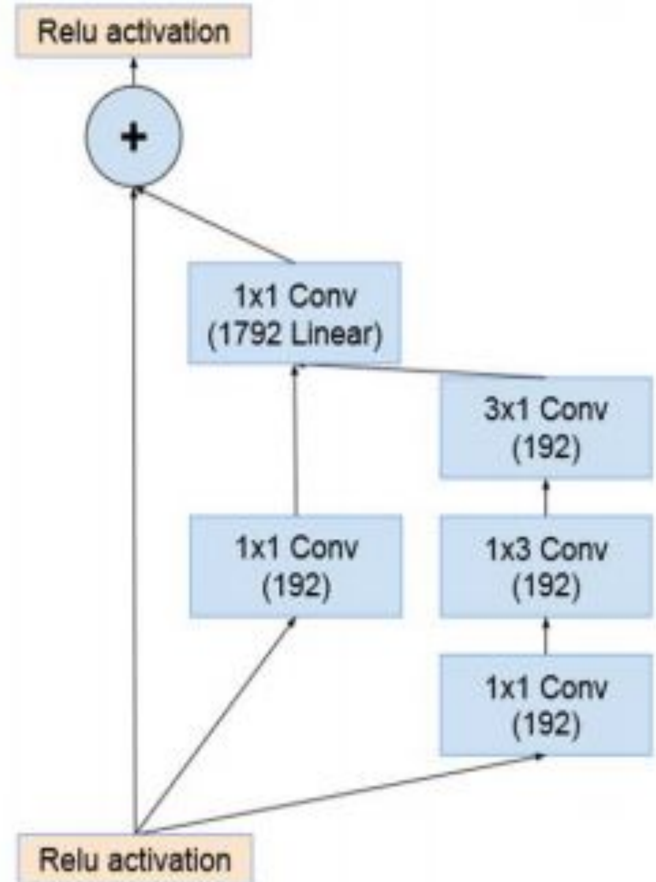
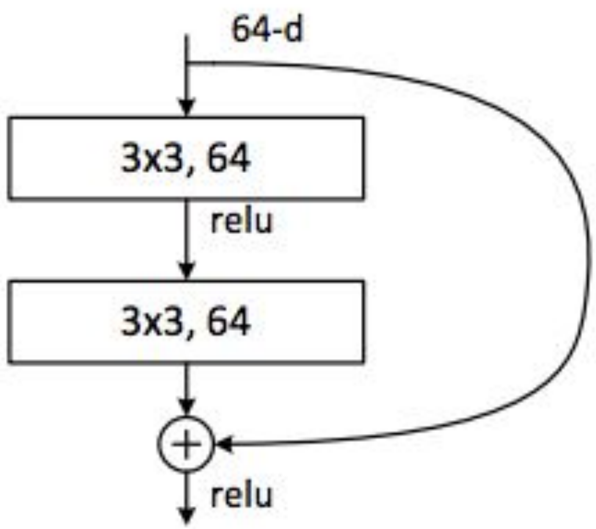


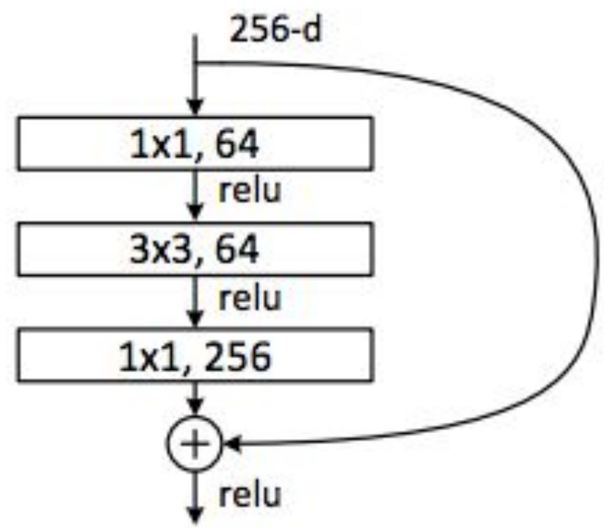
Figure 10. The schema for 35 × 35 grid (Inception-ResNet-A) module of Inception-ResNet-v1 network.



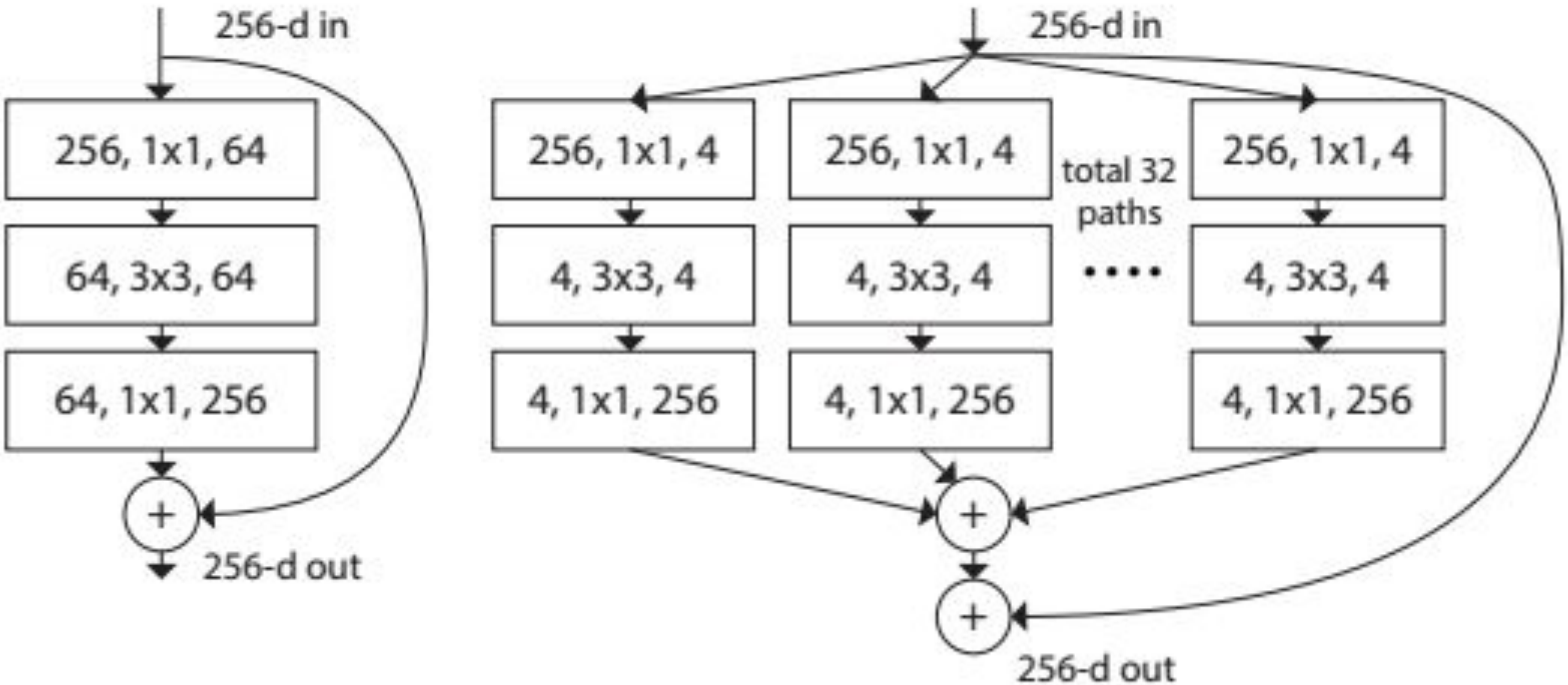
Модуль Inception-ResNet-C для Inception-ResNet-v1

Wide ResNet


64-d
3x3, 64
relu
3x3, 64
relu


256-d
1x1, 64
relu
3x3, 64
relu
1x1, 256
relu

ResNeXt



256-d in
256, 1x1, 64
64, 3x3, 64
64, 1x1, 256
+
256-d out
256-d in
256, 1x1, 4
4, 3x3, 4
4, 1x1, 256
256, 1x1, 4
4, 3x3, 4
4, 1x1, 256
total 32 paths
....
256, 1x1, 4
4, 3x3, 4
4, 1x1, 256
+
+
256-d out

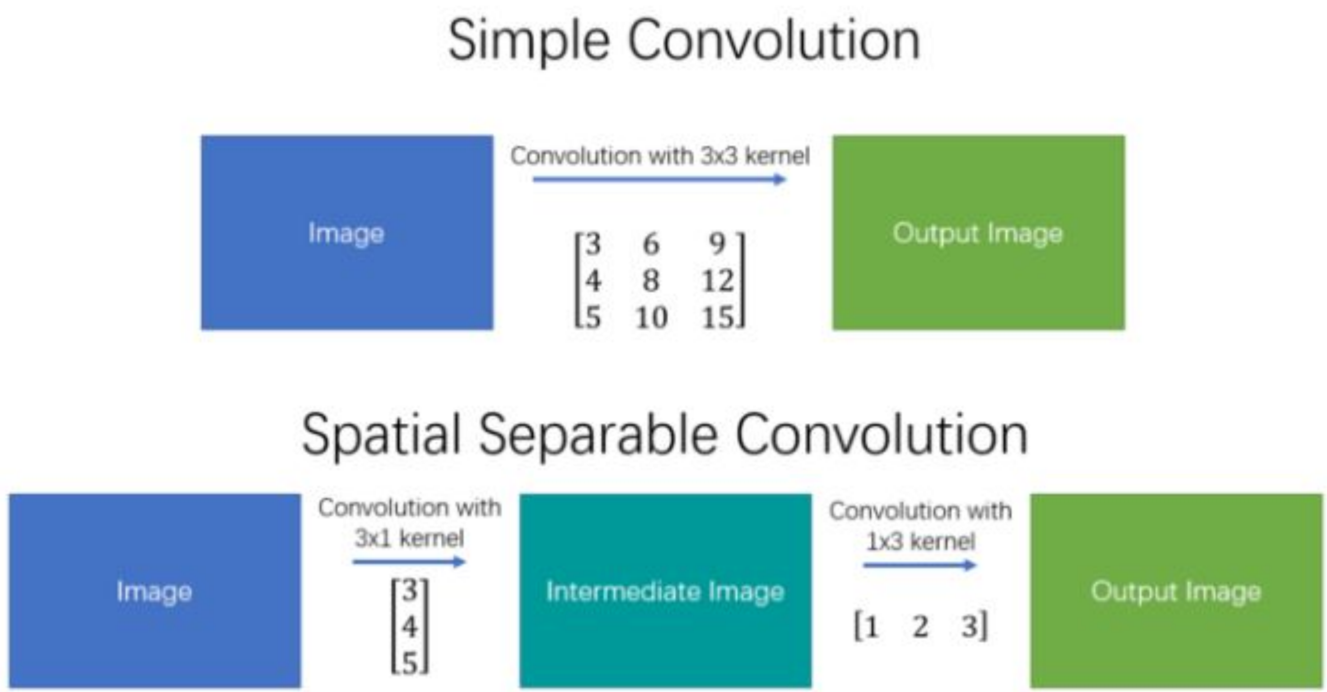
Нащо нам пропускні з'єднання?

- Допомогти мережі вивчити тотожнє перетворення
- Передати ваги вперед в мережі
- Щоб збільшити складність в мережі
- Додати нелінійності

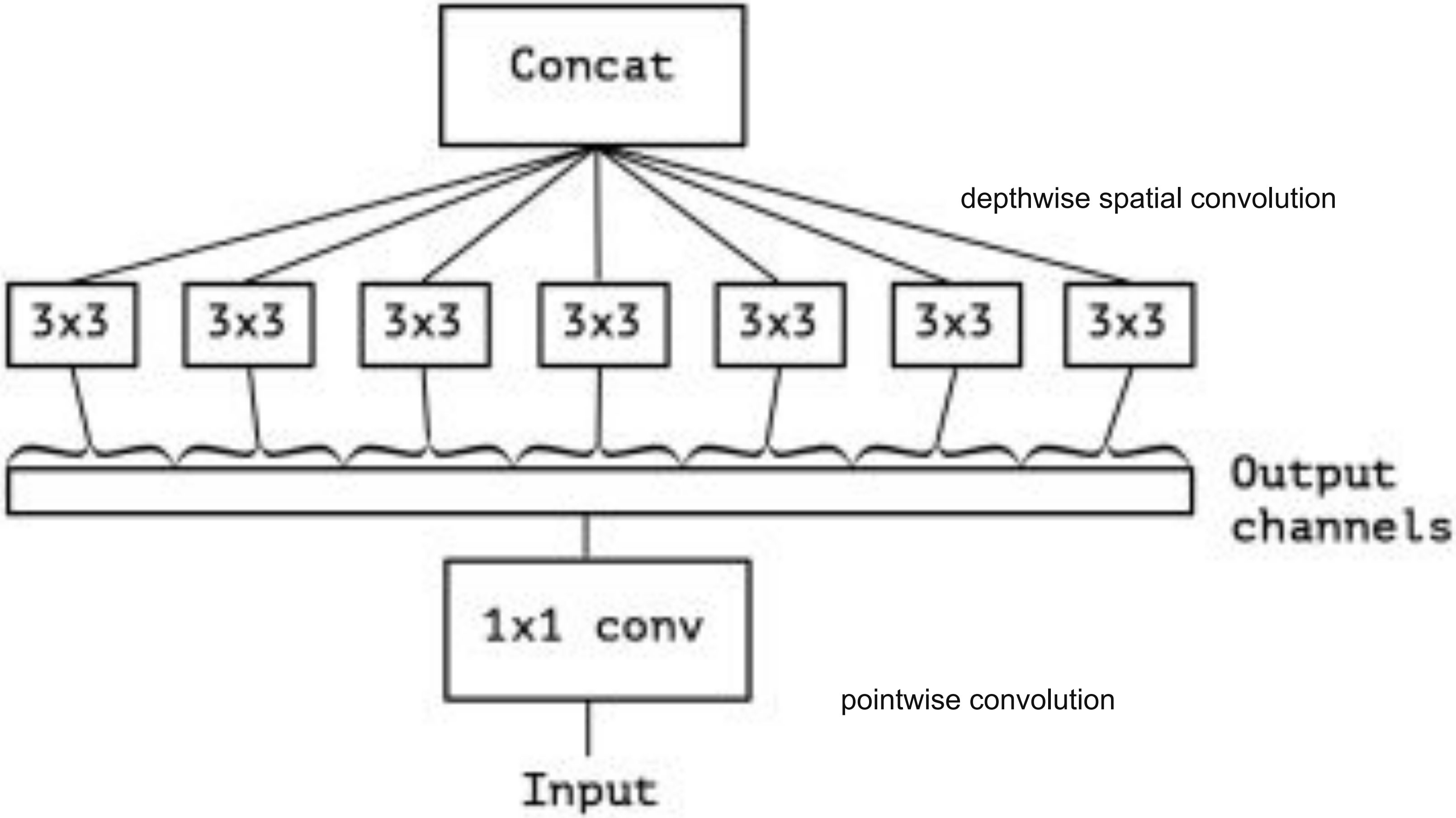
Spatial Separable Convolutions

- Ядро 3x3 можна замінити на 2 ядра 3x1 і 1x3

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



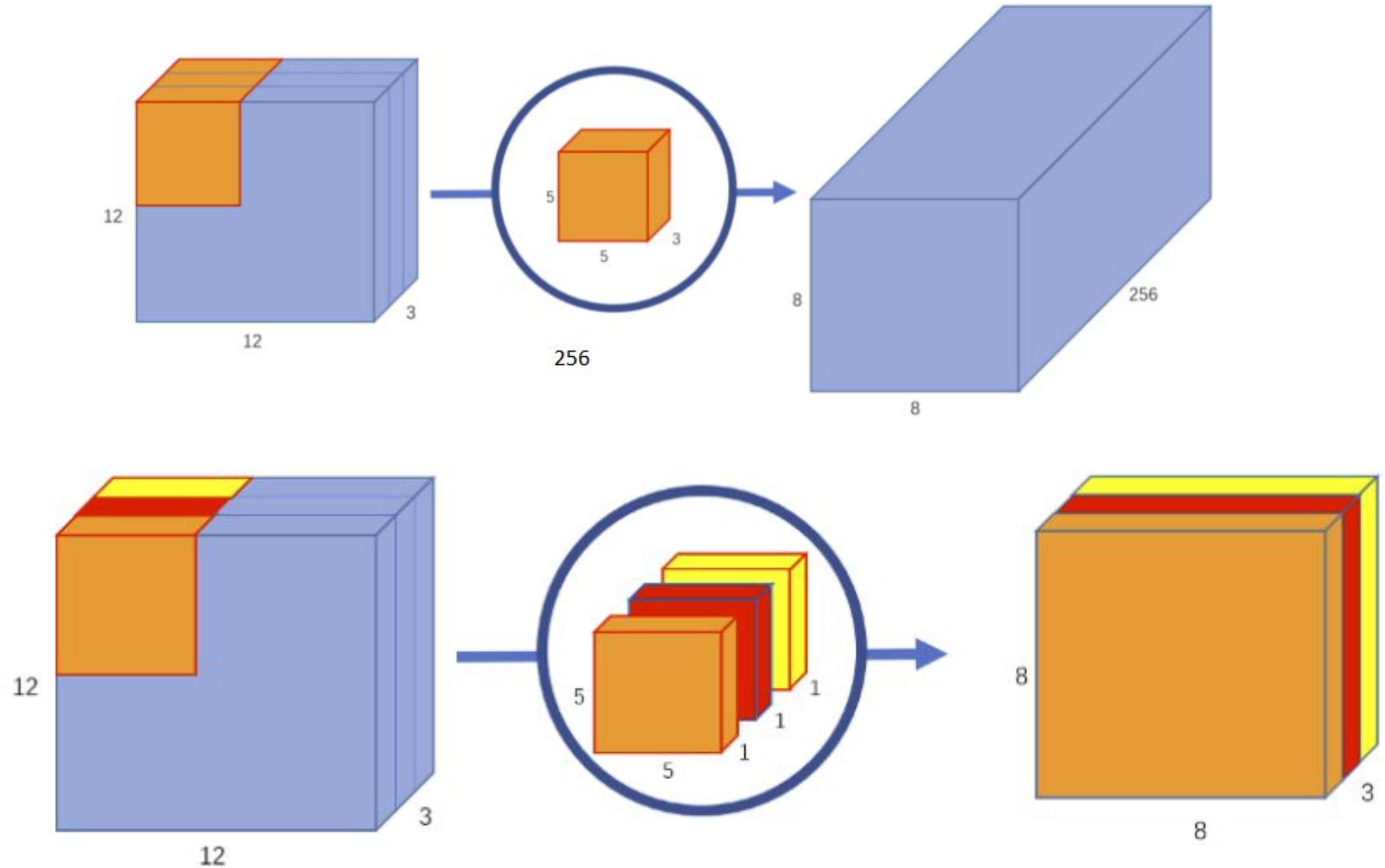
Xception



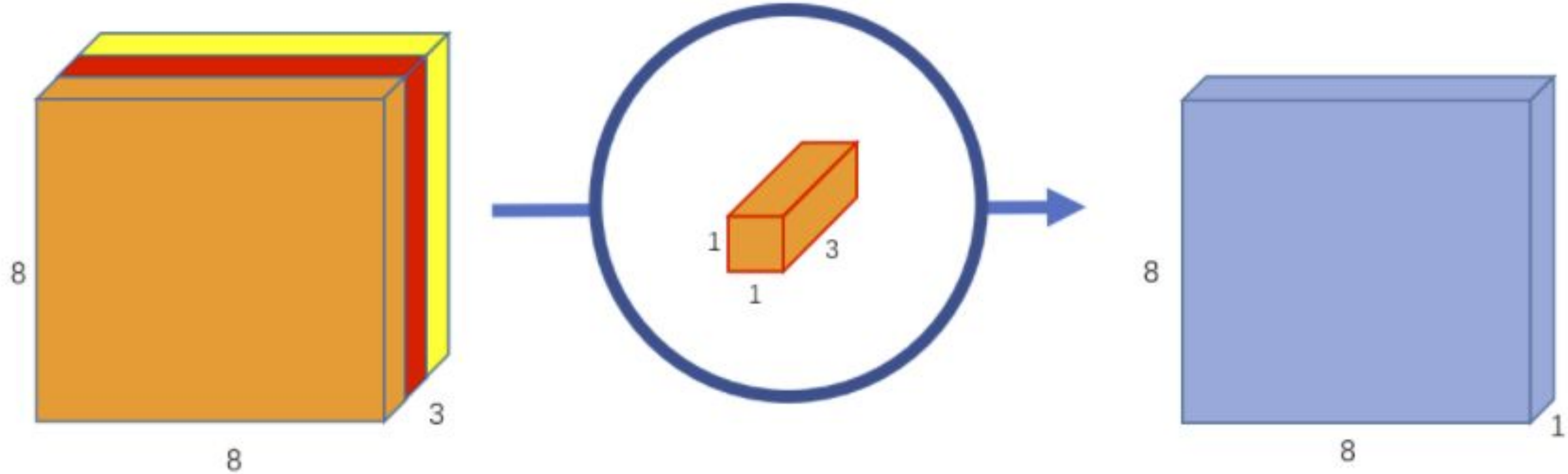
depthwise spatial convolution

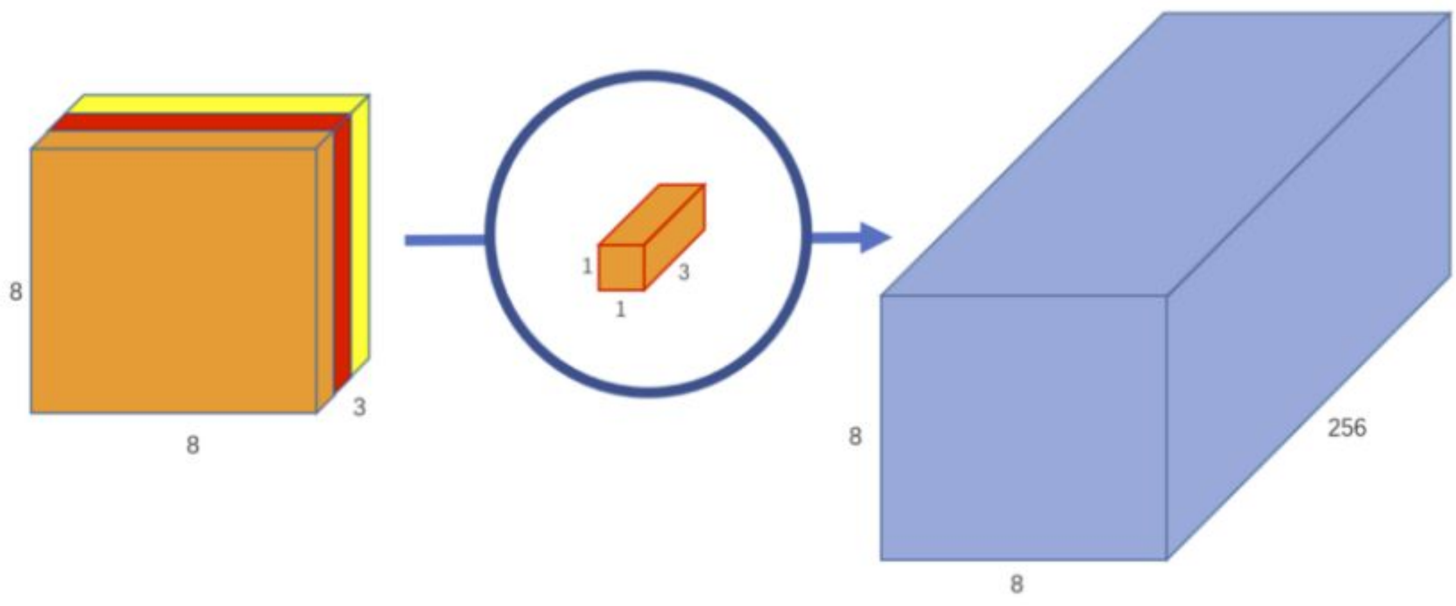
pointwise convolution

Depthwise Convolution

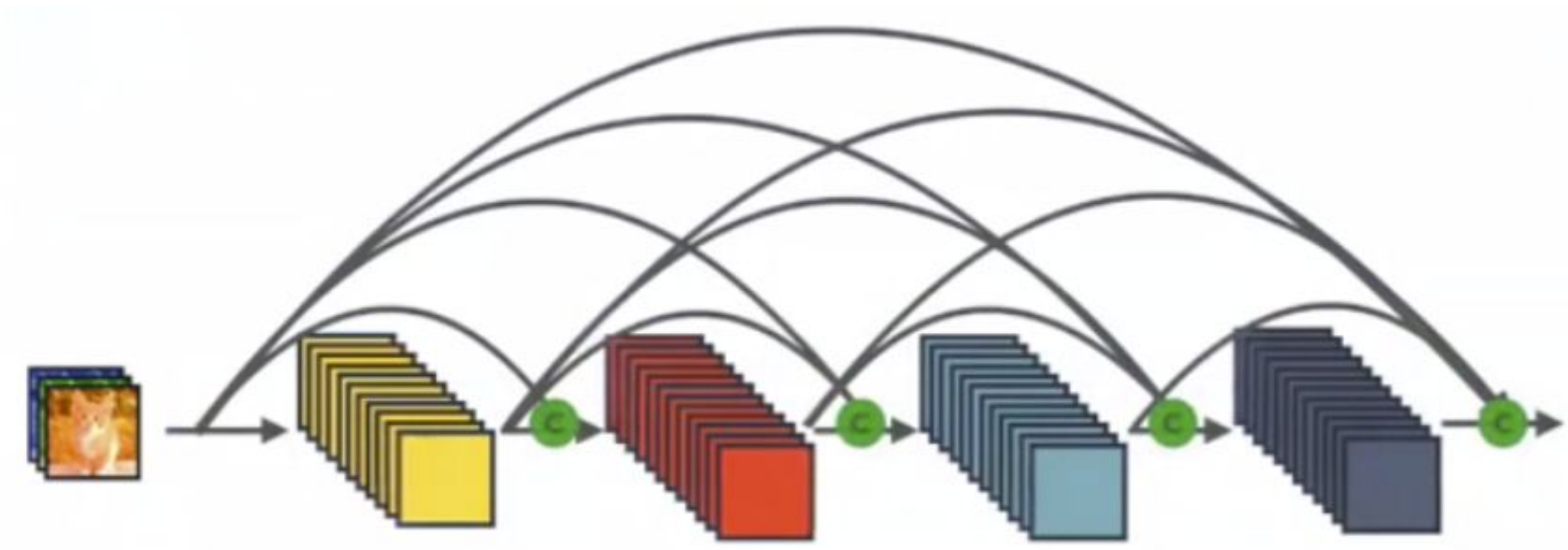


Pointwise Convolution





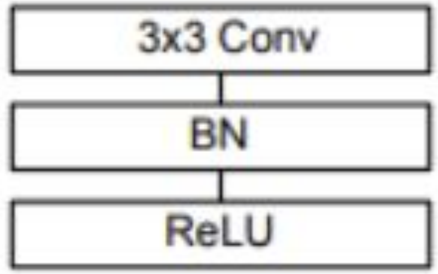
DenseNet

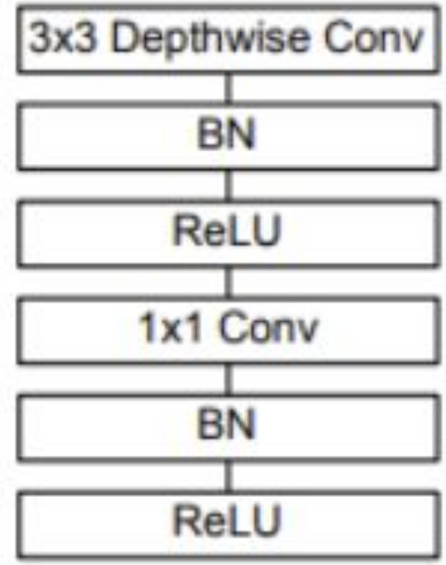


MobileNet

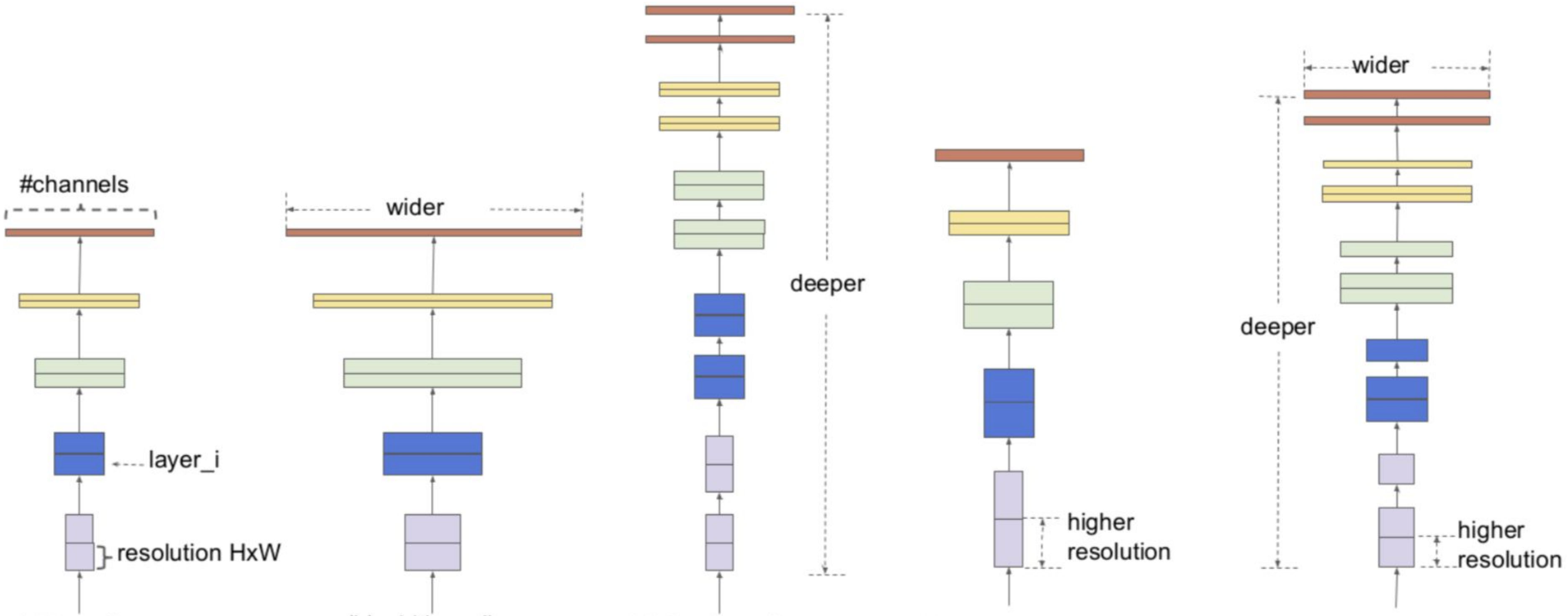
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1 Conv / s1	$3 \times 3 \times 512$ dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$





EfficientNet



(a) baseline (b) width scaling (c) depth scaling (d) resolution scaling (e) compound scaling

EfficientNet

$$\mathcal{N} = \bigodot_{i=1...s} \mathcal{F}_i^{L_i}\big(X_{\langle H_i, W_i, C_i \rangle}\big)$$

$$\mathcal{N}(d, w, r) = \bigodot_{i=1...s} \hat{\mathcal{F}}_i^{d \cdot \hat{L}_i}\big(X_{\langle r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i \rangle}\big)$$

$$\begin{aligned}
\text{depth: } & d = \alpha^\phi \\
\text{width: } & w = \beta^\phi \\
\text{resolution: } & r = \gamma^\phi \\
\text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\
& \alpha \geq 1, \beta \geq 1, \gamma \geq 1
\end{aligned}$$