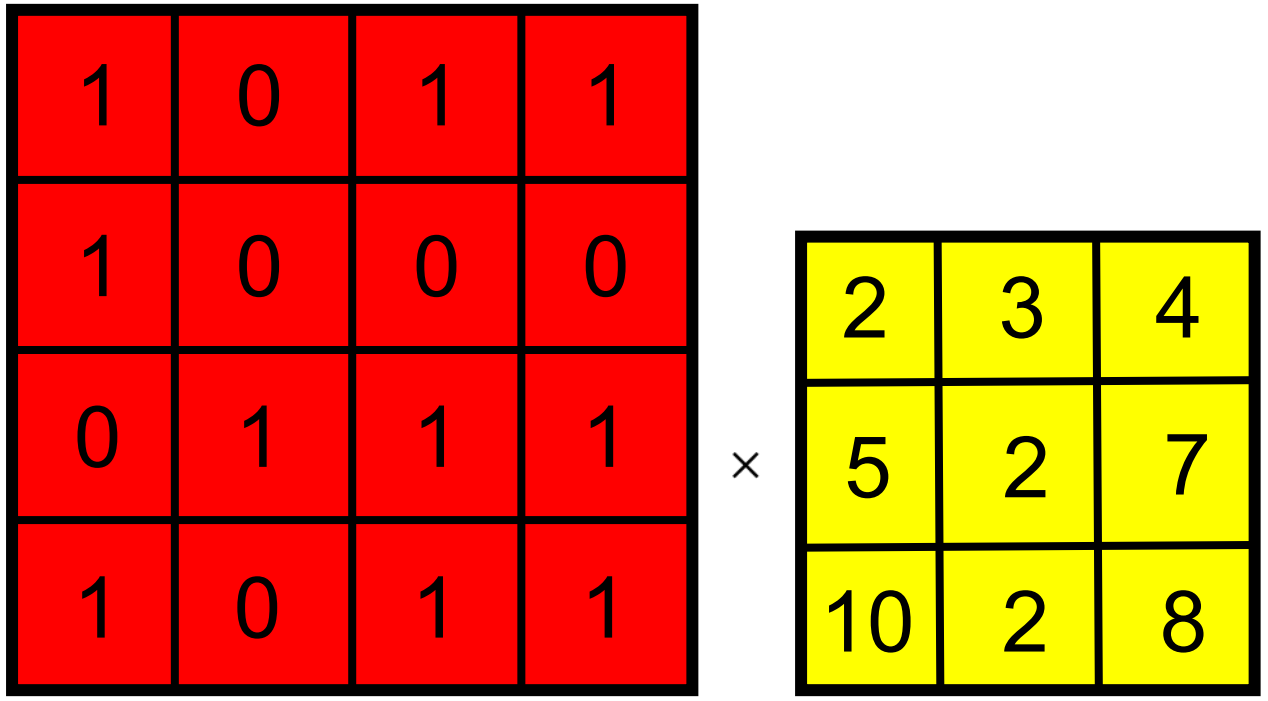
Framework para CNNs

André Zucchetti Dalla Costa

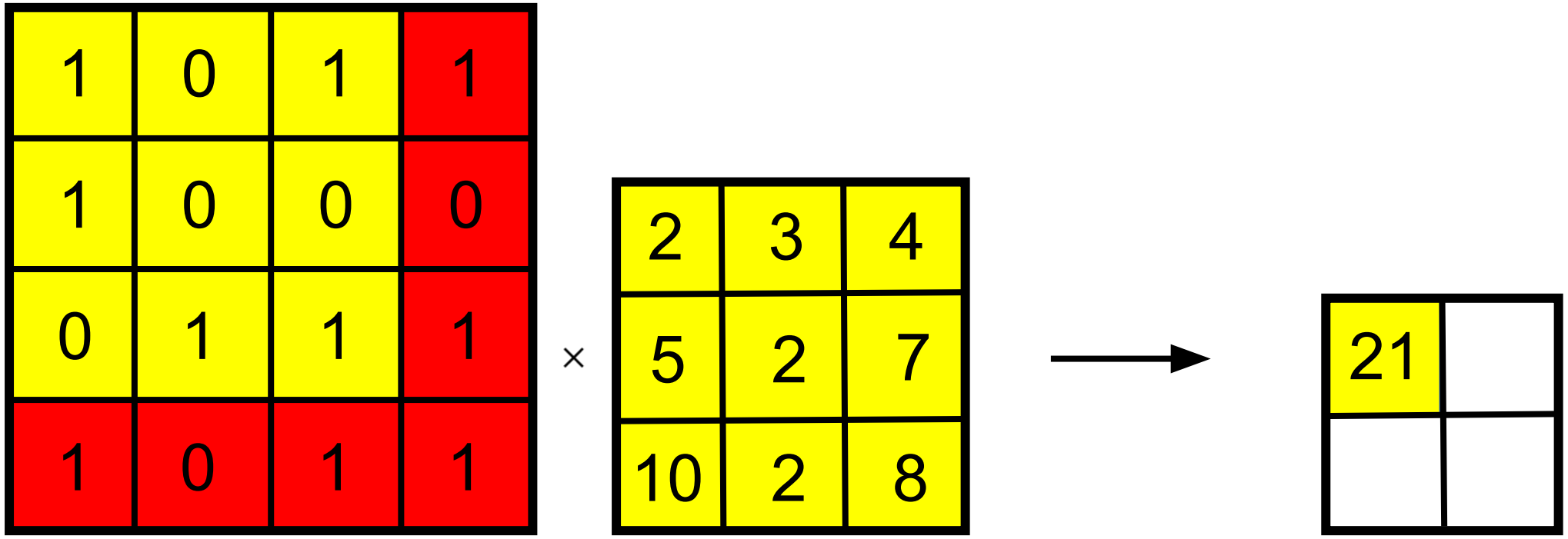
Matheus Corrêa Lindino

Convolution

Convolution → $\sum_{n=0}^{k} f(n) * g(k-n)$ → Kernel = 7x7, 5x5, 3x3
Stride = 1
Padding = 0

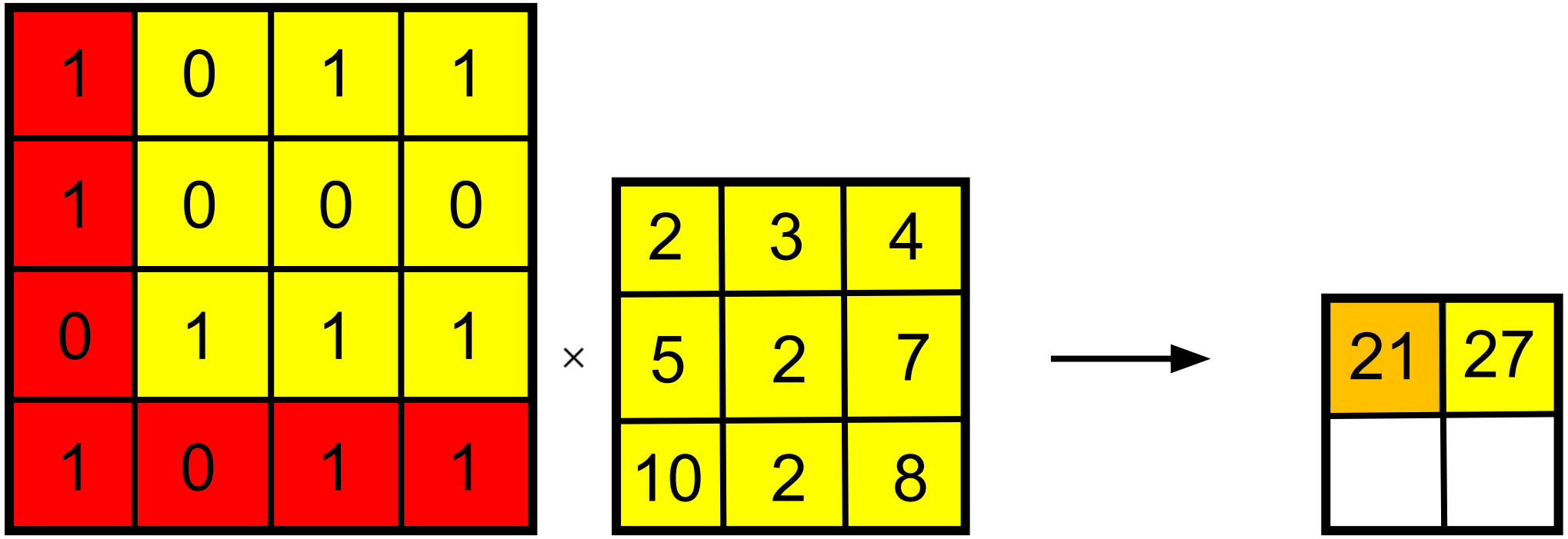


Convolution → $\sum_{n=0}^{k} f(n) * g(k-n)$ → Kernel = 7x7, 5x5, 3x3, Stride = 1, Padding = 0



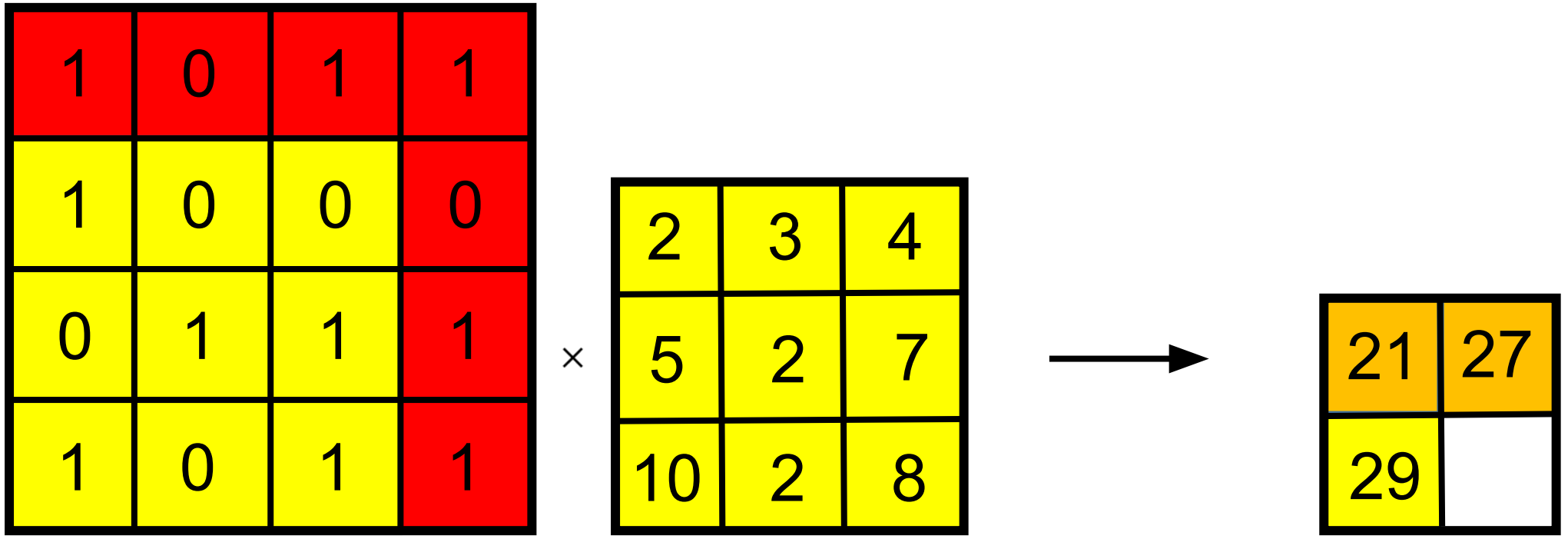
$$(2 * 1) + (3 * 0) + (4 * 1) + (5 * 1) + (2 * 0) + (7 * 0) + (10 * 0) + (2 * 1) + (8 * 1) = 21$$

Convolution → $\sum_{n=0}^{k} f(n) * g(k-n)$ → Kernel = 7x7, 5x5, 3x3, Stride = 1, Padding = 0



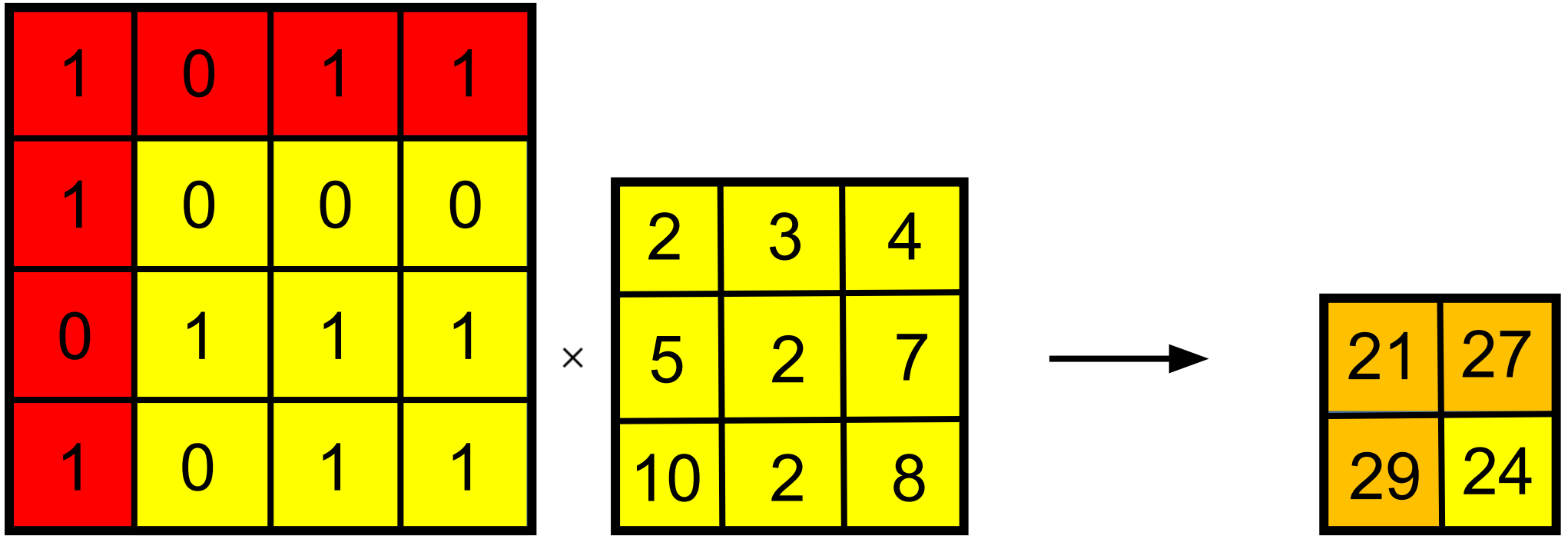
$$(2 * 0) + (3 * 1) + (4 * 1) + (5 * 0) + (2 * 0) + (7 * 0) + (10 * 1) + (2 * 1) + (8 * 1) = 27$$

Convolution → $\sum_{n=0}^{k} f(n) * g(k-n)$ → Kernel = 7x7, 5x5, 3x3, Stride = 1, Padding = 0



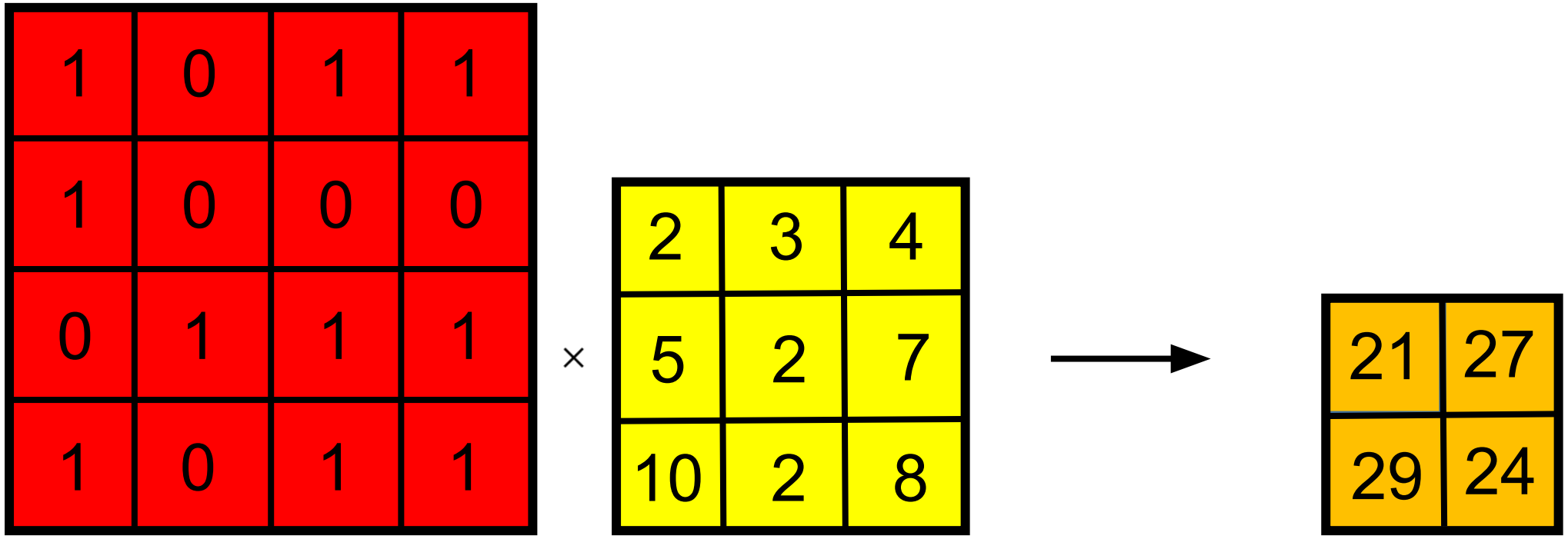
$$(2 * 1) + (3 * 0) + (4 * 0) + (5 * 0) + (2 * 1) + (7 * 1) + (10 * 1) + (2 * 0) + (8 * 1) = 29$$

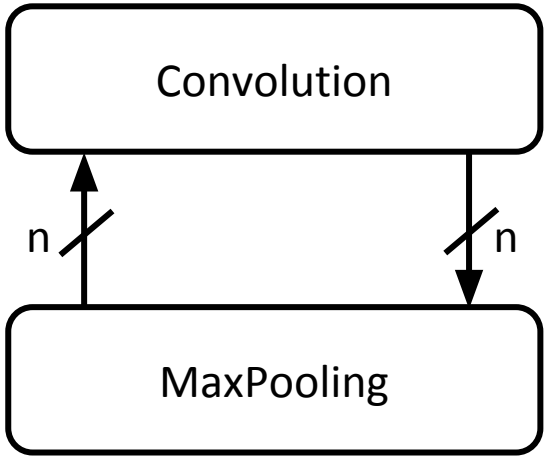
Convolution → $\sum_{n=0}^{k} f(n) * g(k-n)$ → Kernel = 7x7, 5x5, 3x3, Stride = 1, Padding = 0

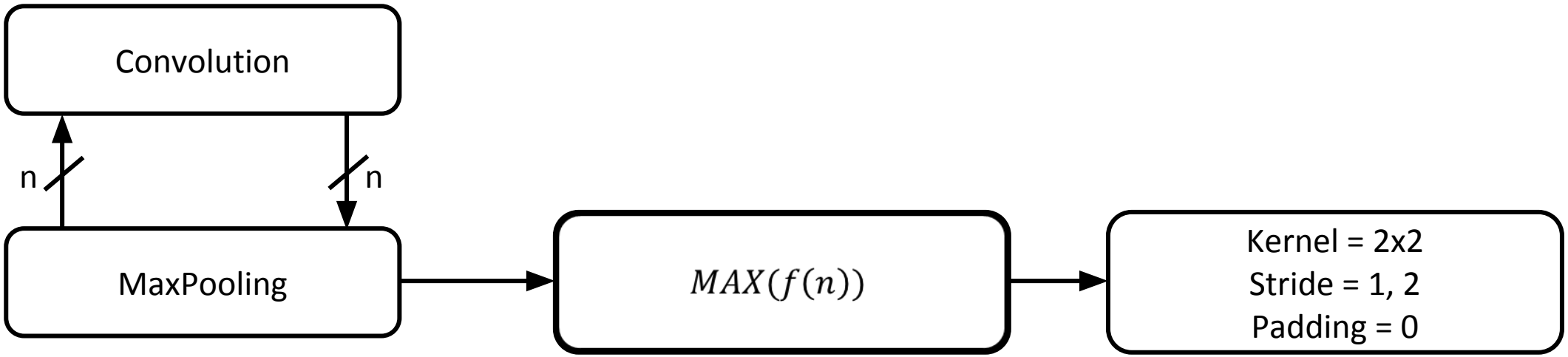


$$(2 * 0) + (3 * 0) + (4 * 0) + (5 * 1) + (2 * 1) + (7 * 1) + (10 * 0) + (2 * 1) + (8 * 1) = 24$$

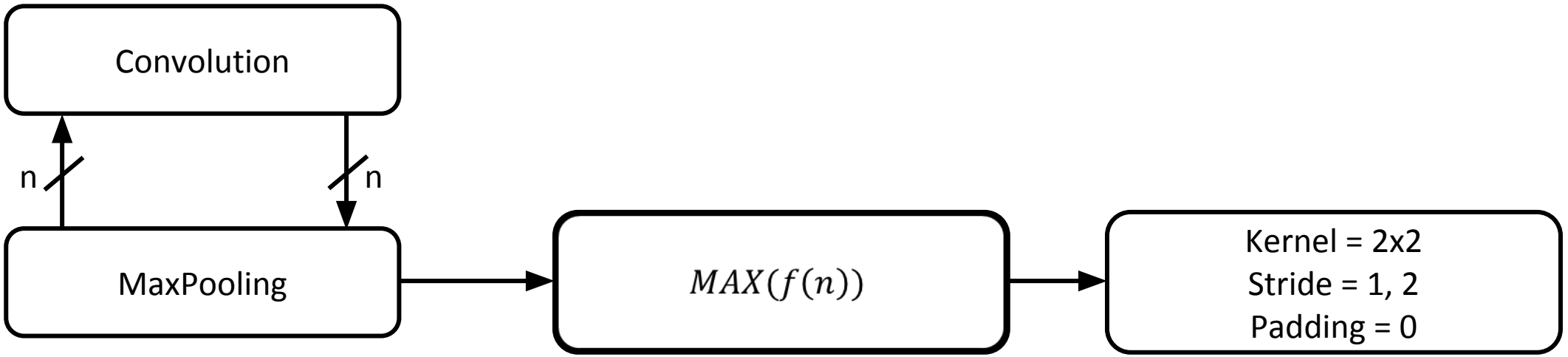
Convolution → $\sum_{n=0}^{k} f(n) * g(k-n)$ → Kernel = 7x7, 5x5, 3x3
Stride = 1
Padding = 0

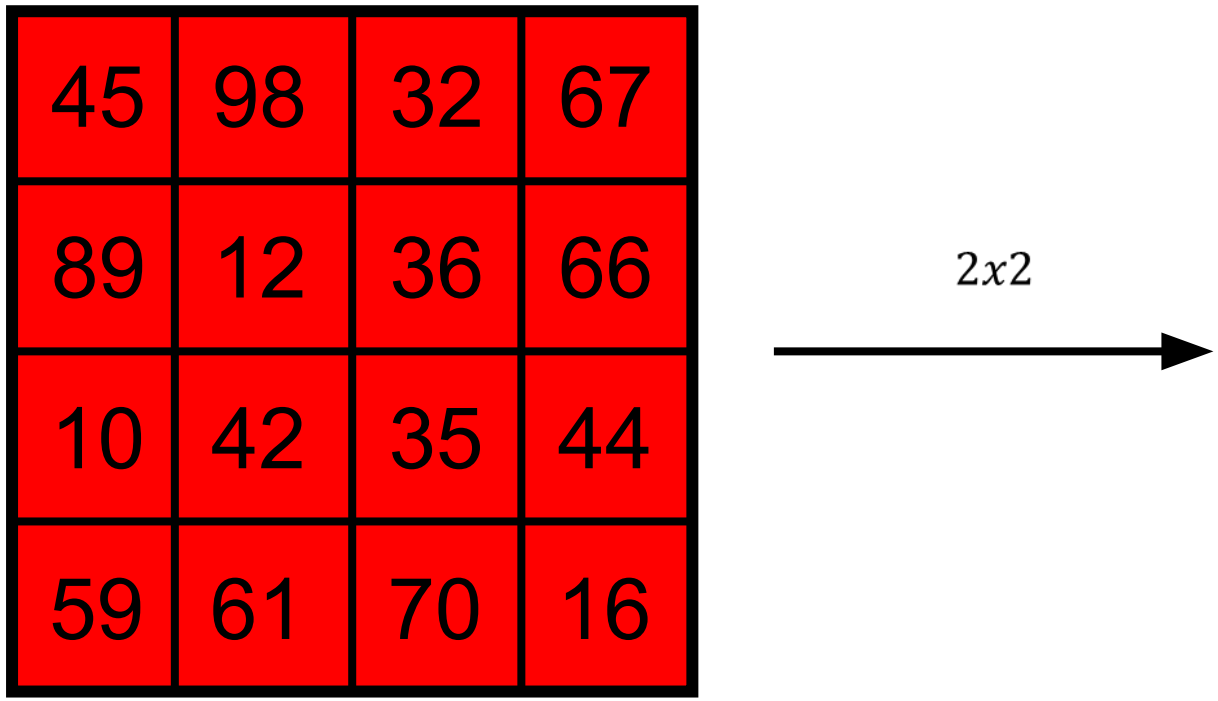


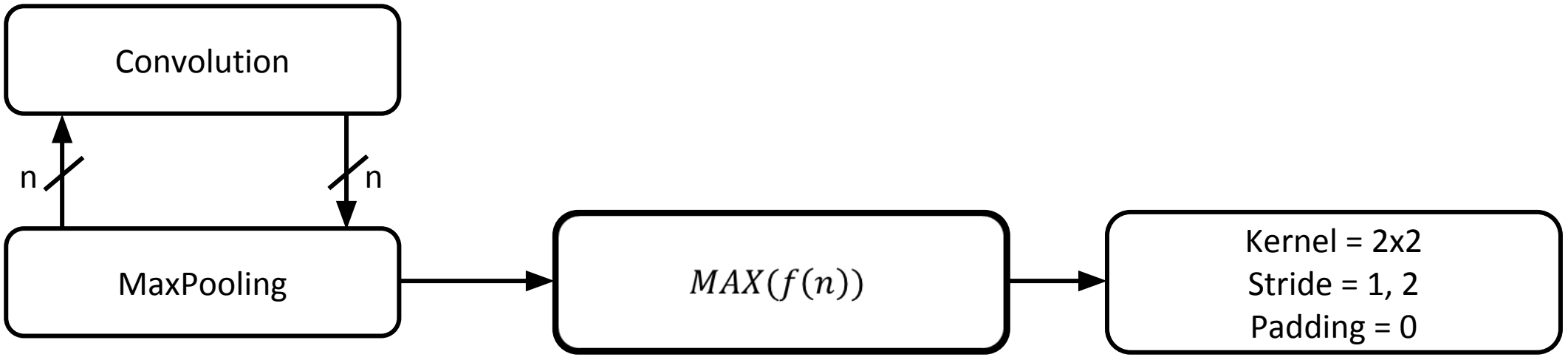

Convolution
n
n
MaxPooling

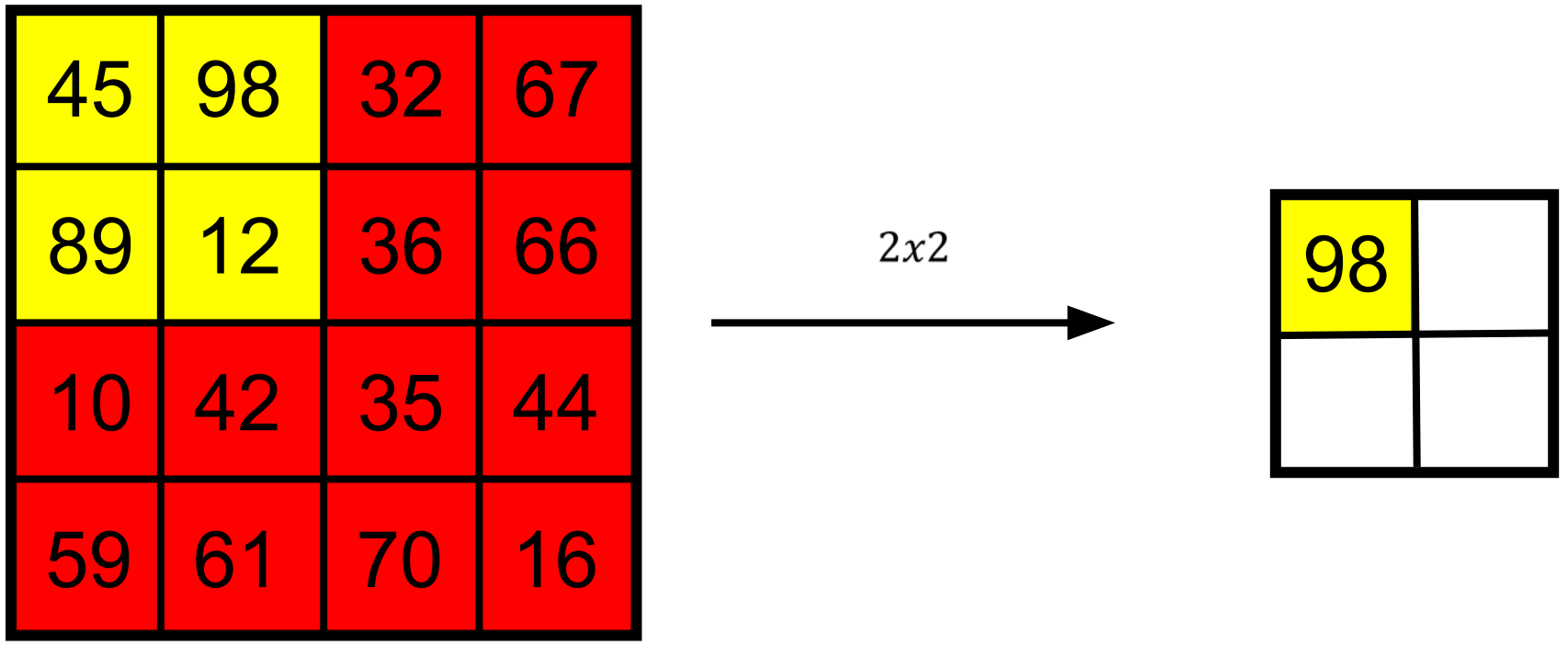


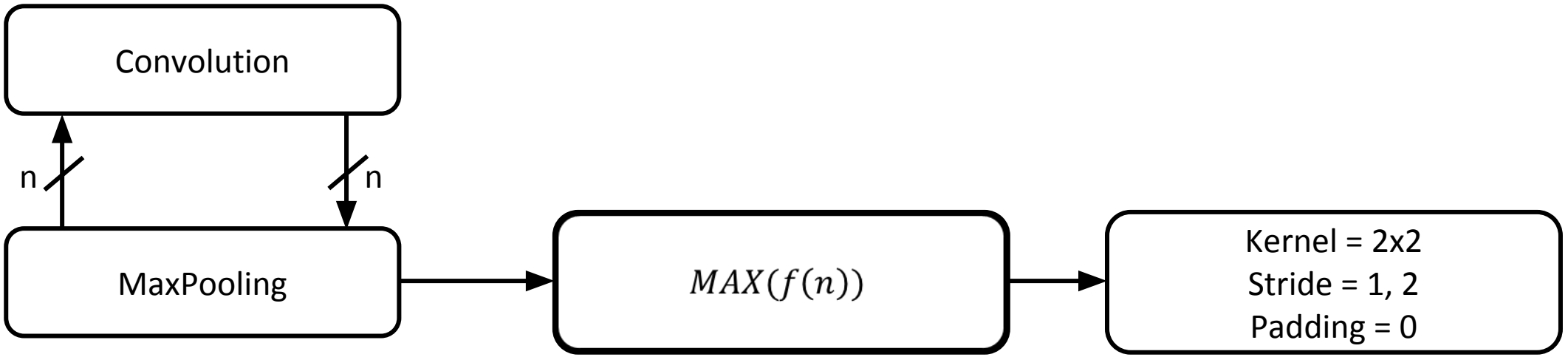
45	98	32	67
89	12	36	66
10	42	35	44
59	61	70	16

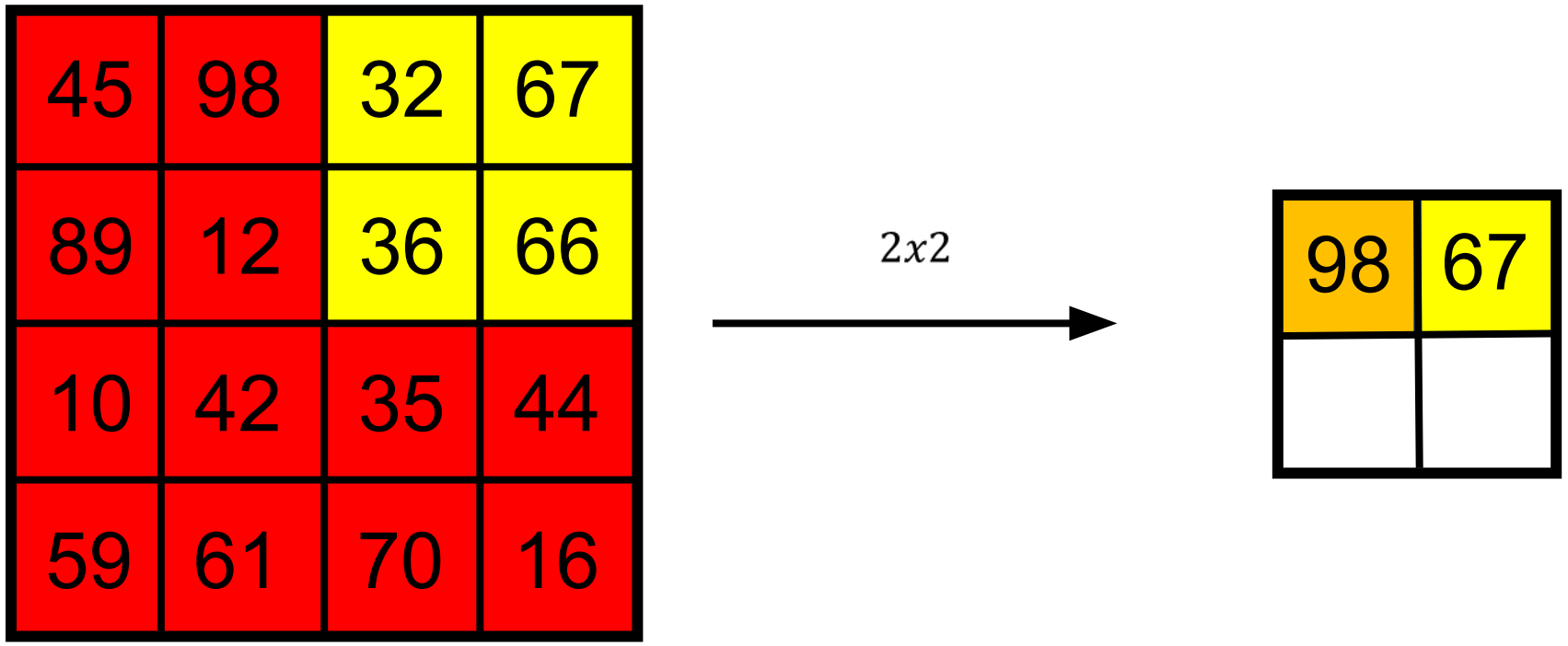

Convolution
n
n
MaxPooling
$MAX(f(n))$
Kernel = 2x2
Stride = 1, 2
Padding = 0

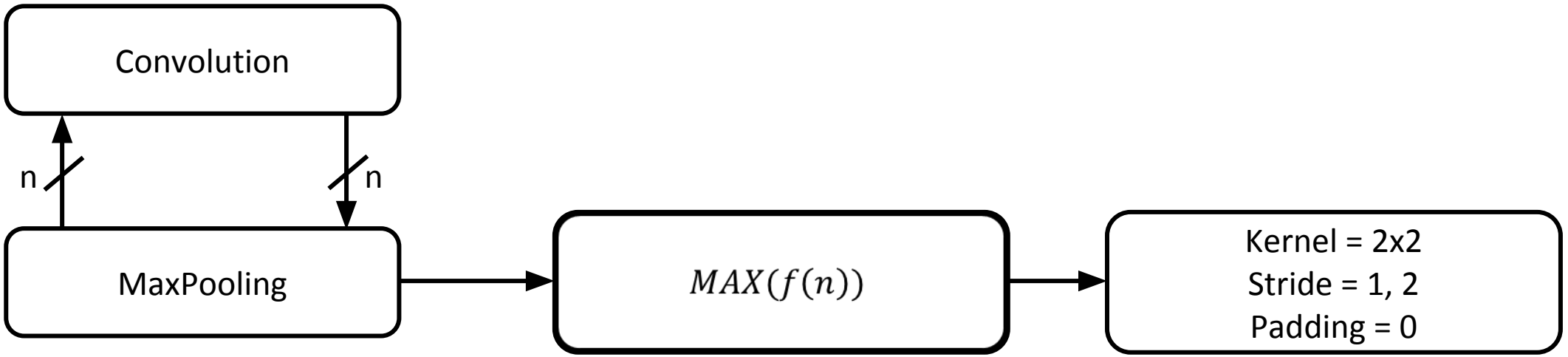

45 98 32 67
89 12 36 66
10 42 35 44
59 61 70 16
$2x2$

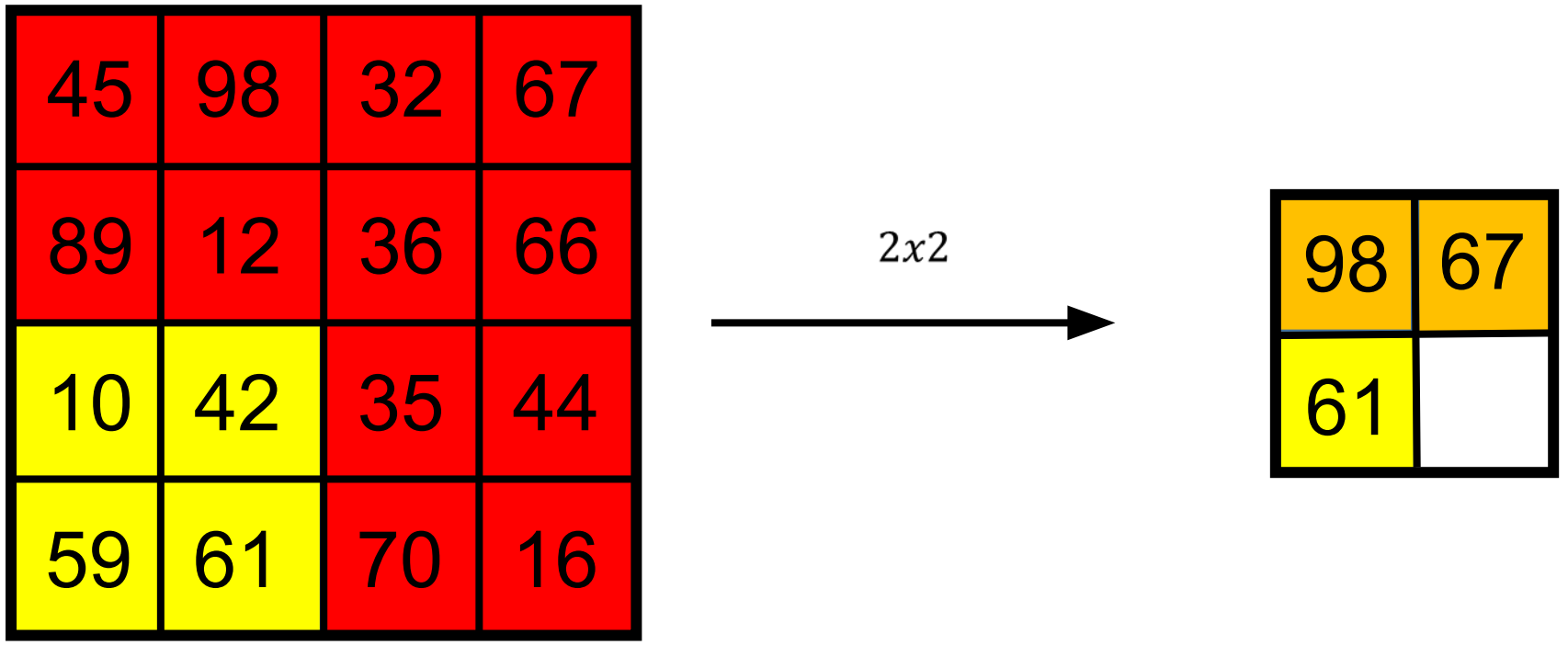

Convolution
n
n
MaxPooling
$MAX(f(n))$
Kernel = 2x2
Stride = 1, 2
Padding = 0

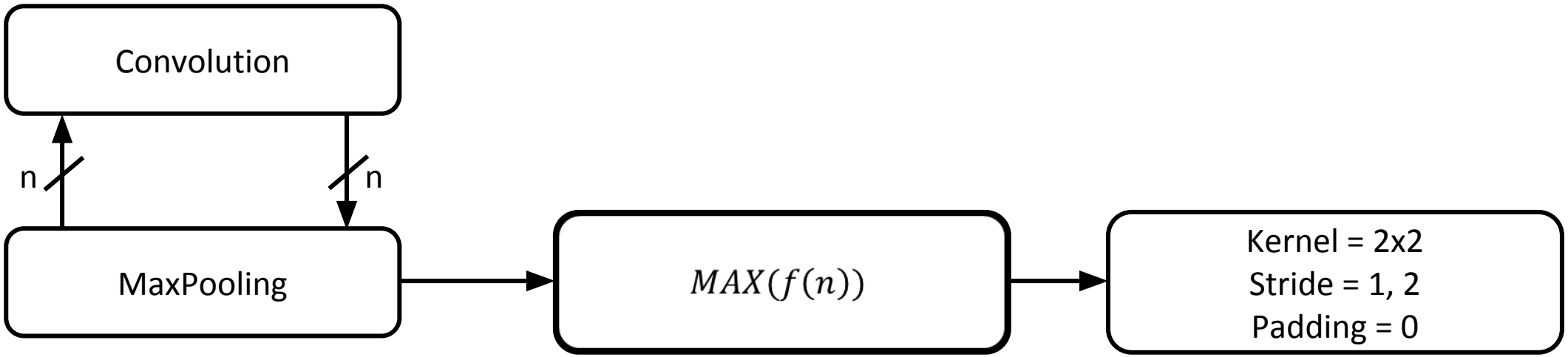

45 98 32 67
89 12 36 66
10 42 35 44
59 61 70 16
$2x2$
98

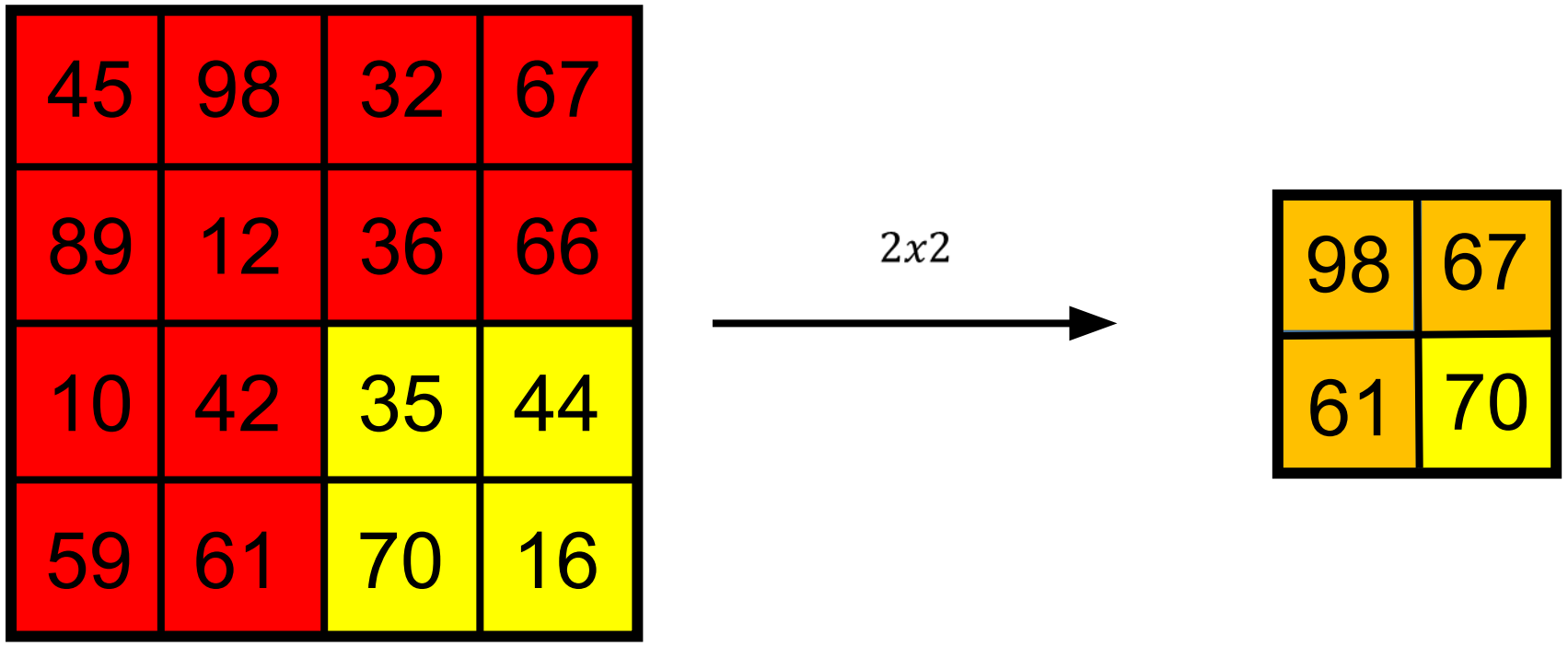

Convolution
n
n
MaxPooling
$MAX(f(n))$
Kernel = 2x2
Stride = 1, 2
Padding = 0

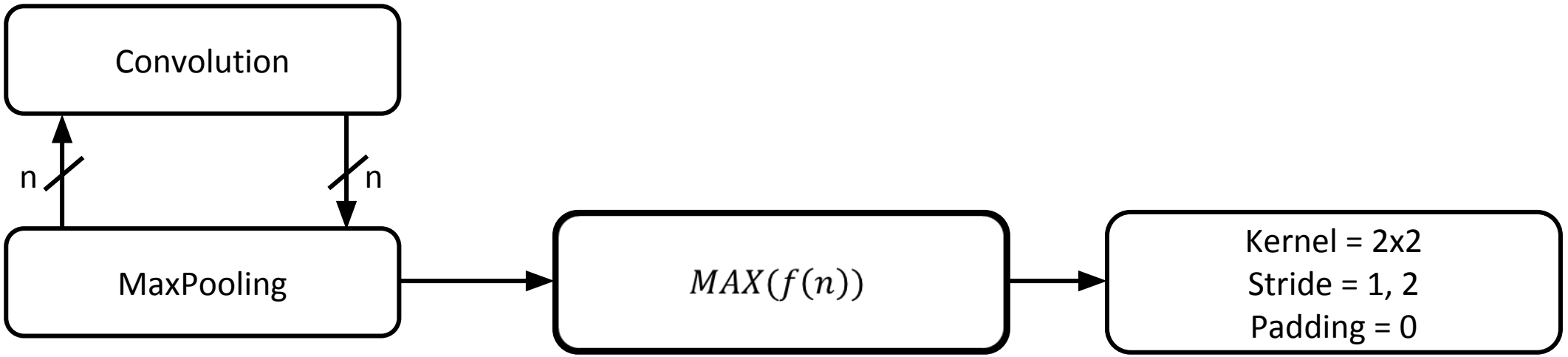

45 98 32 67
89 12 36 66
10 42 35 44
59 61 70 16
$2x2$
98 67

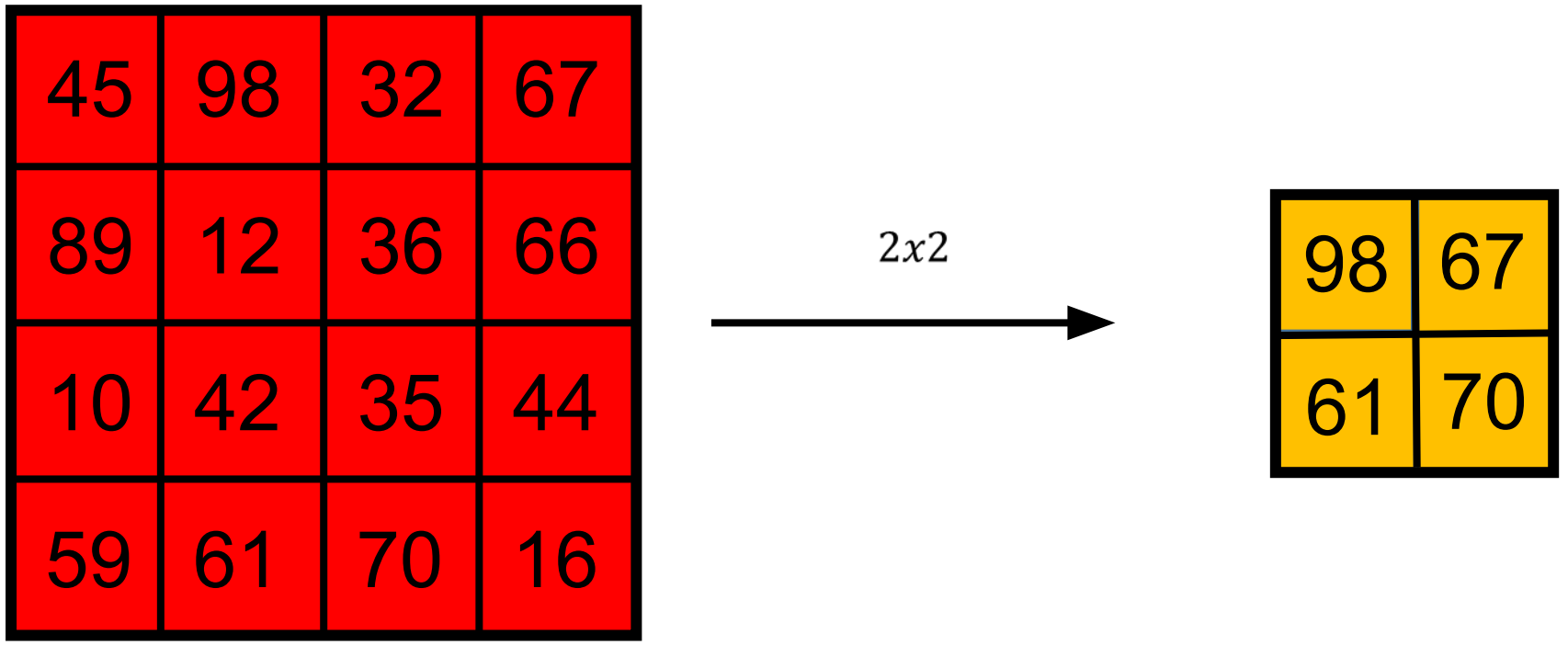

Convolution
n
n
MaxPooling
$MAX(f(n))$
Kernel = 2x2
Stride = 1, 2
Padding = 0

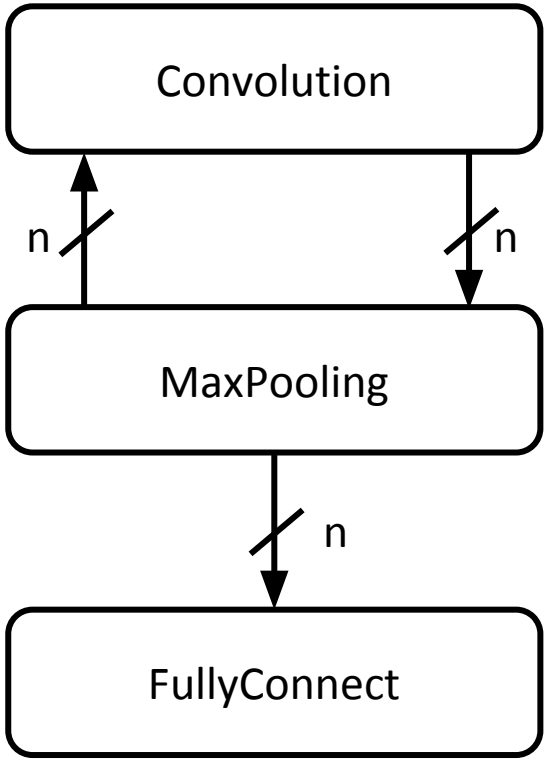

45 98 32 67
89 12 36 66
10 42 35 44
59 61 70 16
$2x2$
98 67
61

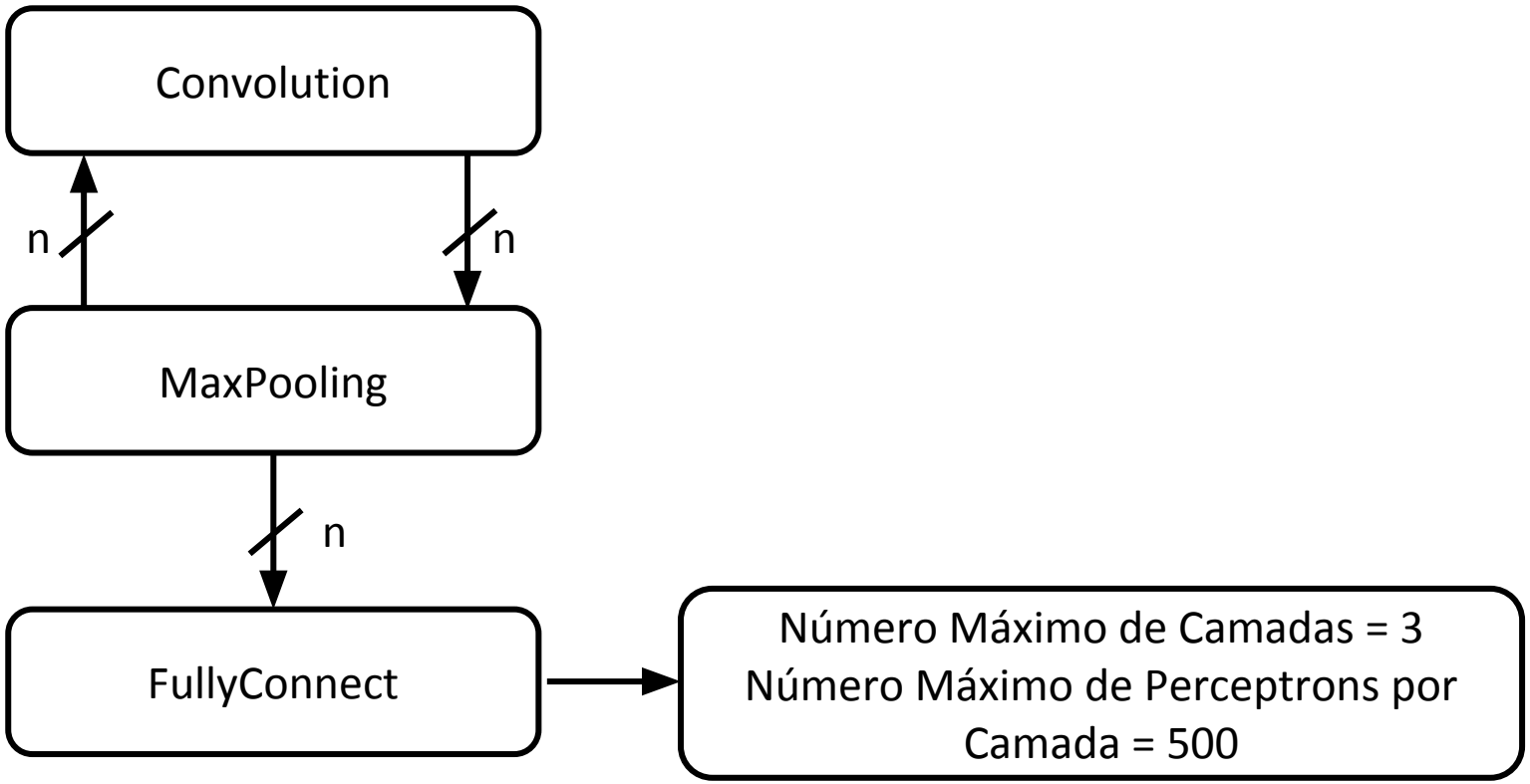

Convolution
n
n
MaxPooling
$MAX(f(n))$
Kernel = 2x2
Stride = 1, 2
Padding = 0

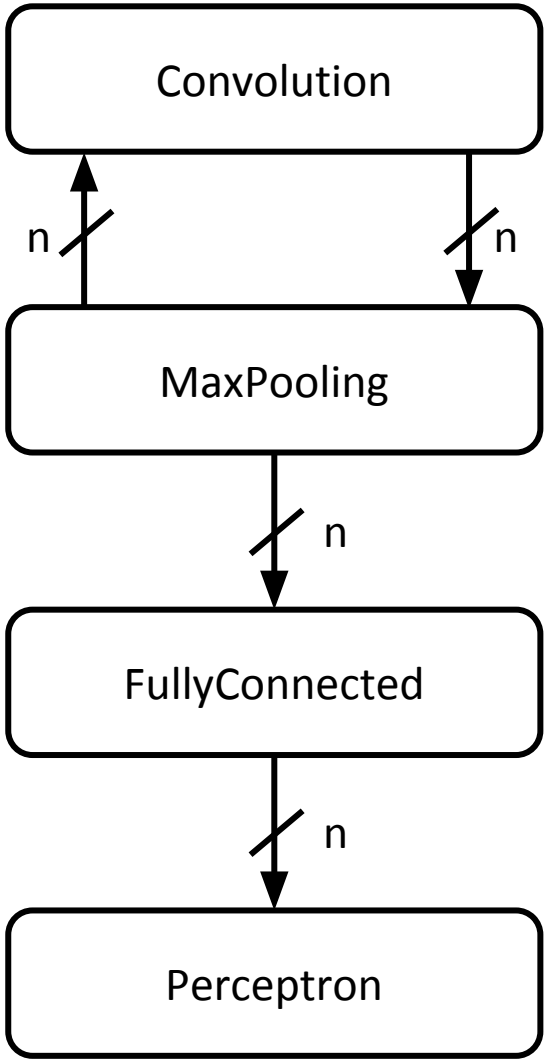

45 98 32 67
89 12 36 66
10 42 35 44
59 61 70 16
$2x2$
98 67
61 70

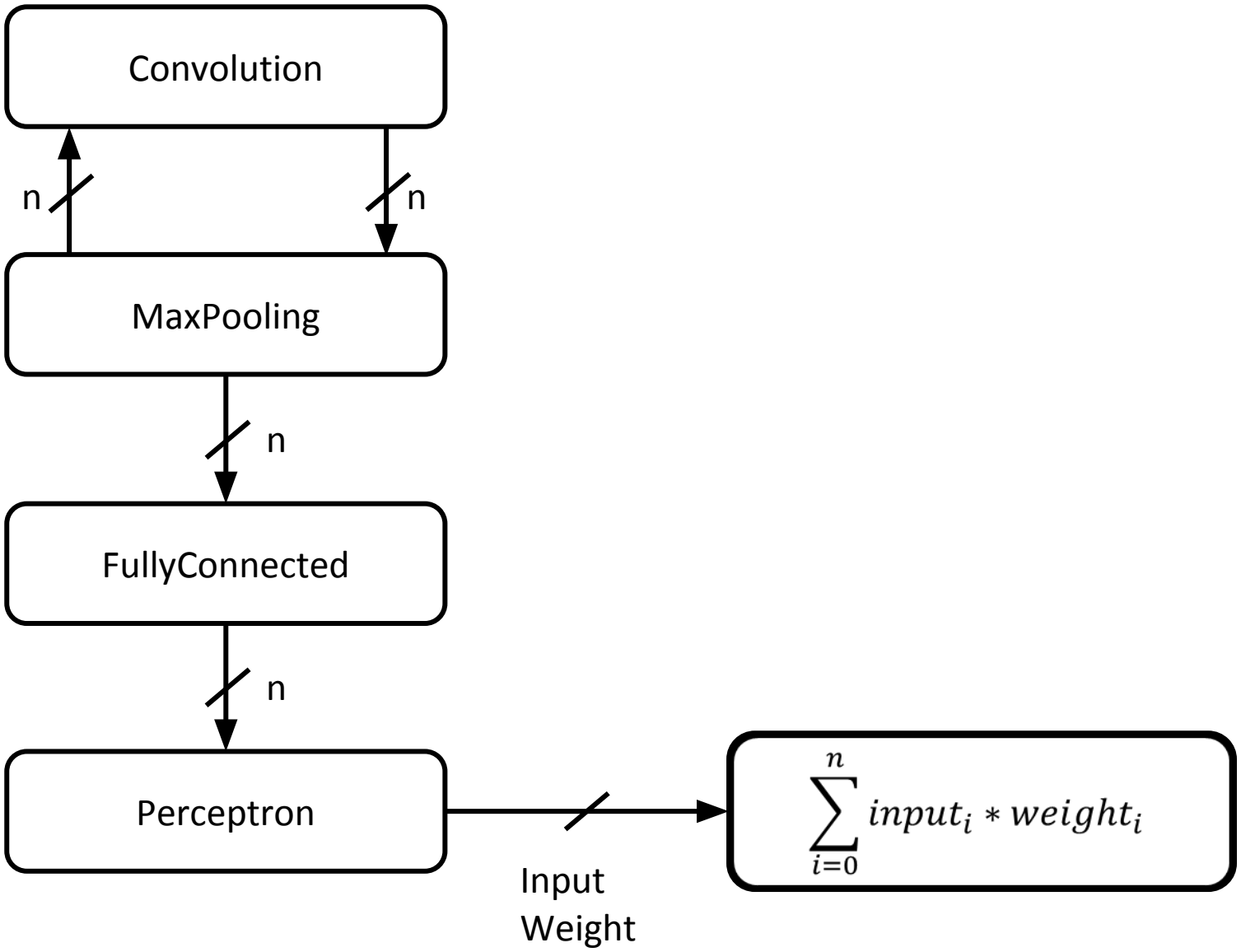

Convolution
n
n
MaxPooling
$MAX(f(n))$
Kernel = 2x2
Stride = 1, 2
Padding = 0

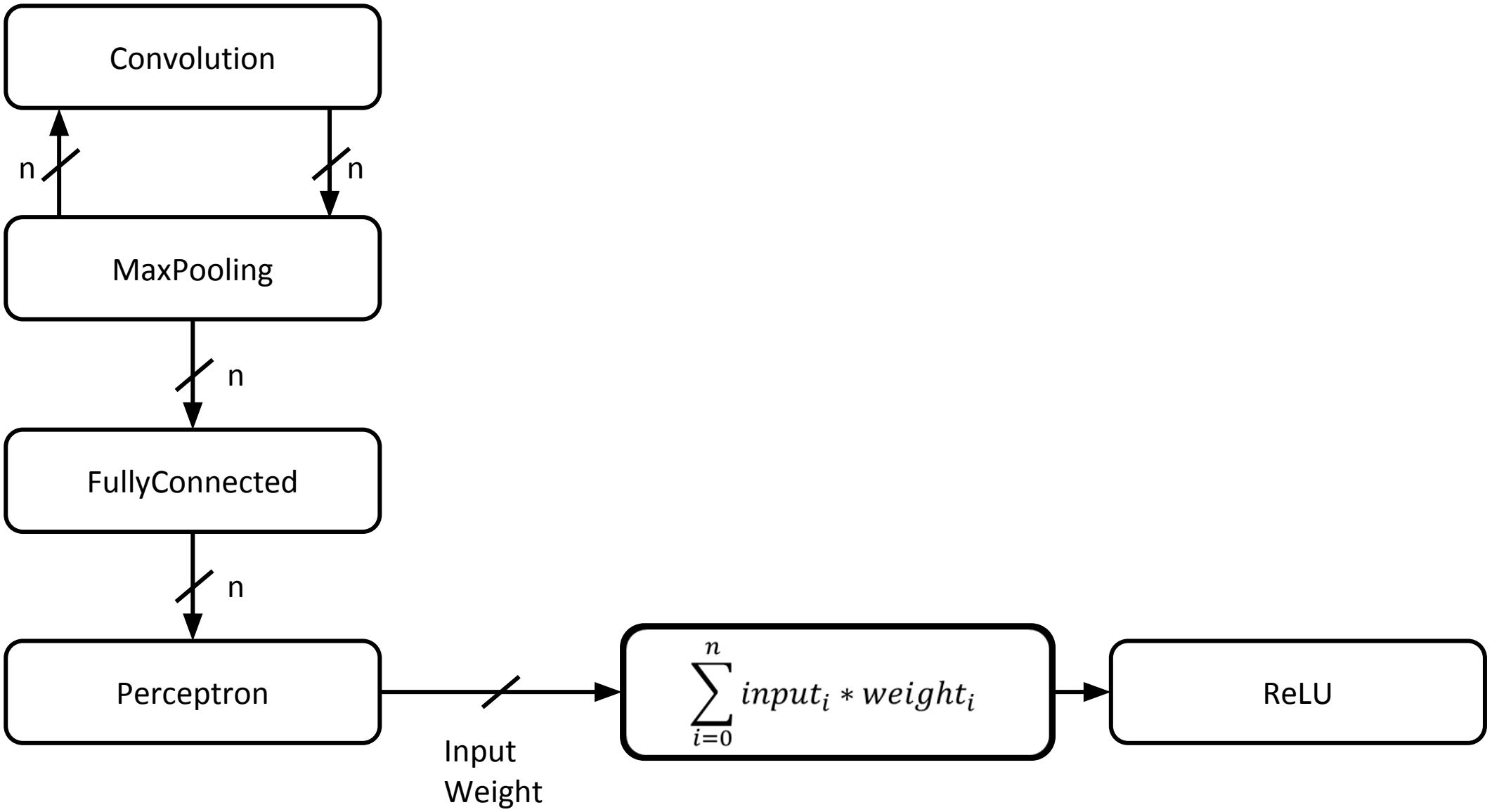

45 98 32 67
89 12 36 66
10 42 35 44
59 61 70 16
$2x2$
98 67
61 70

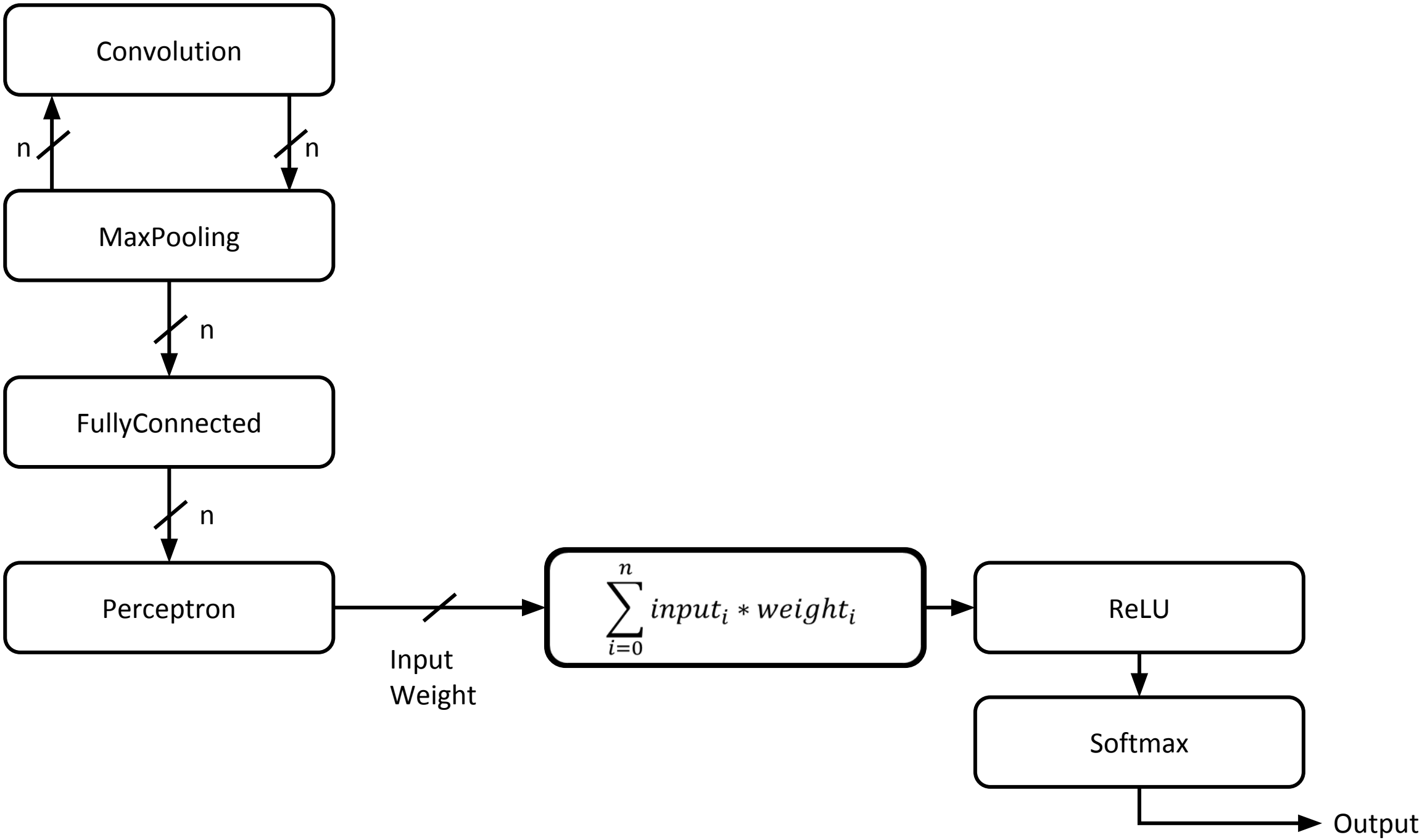

Convolution
n
n
MaxPooling
n
FullyConnect

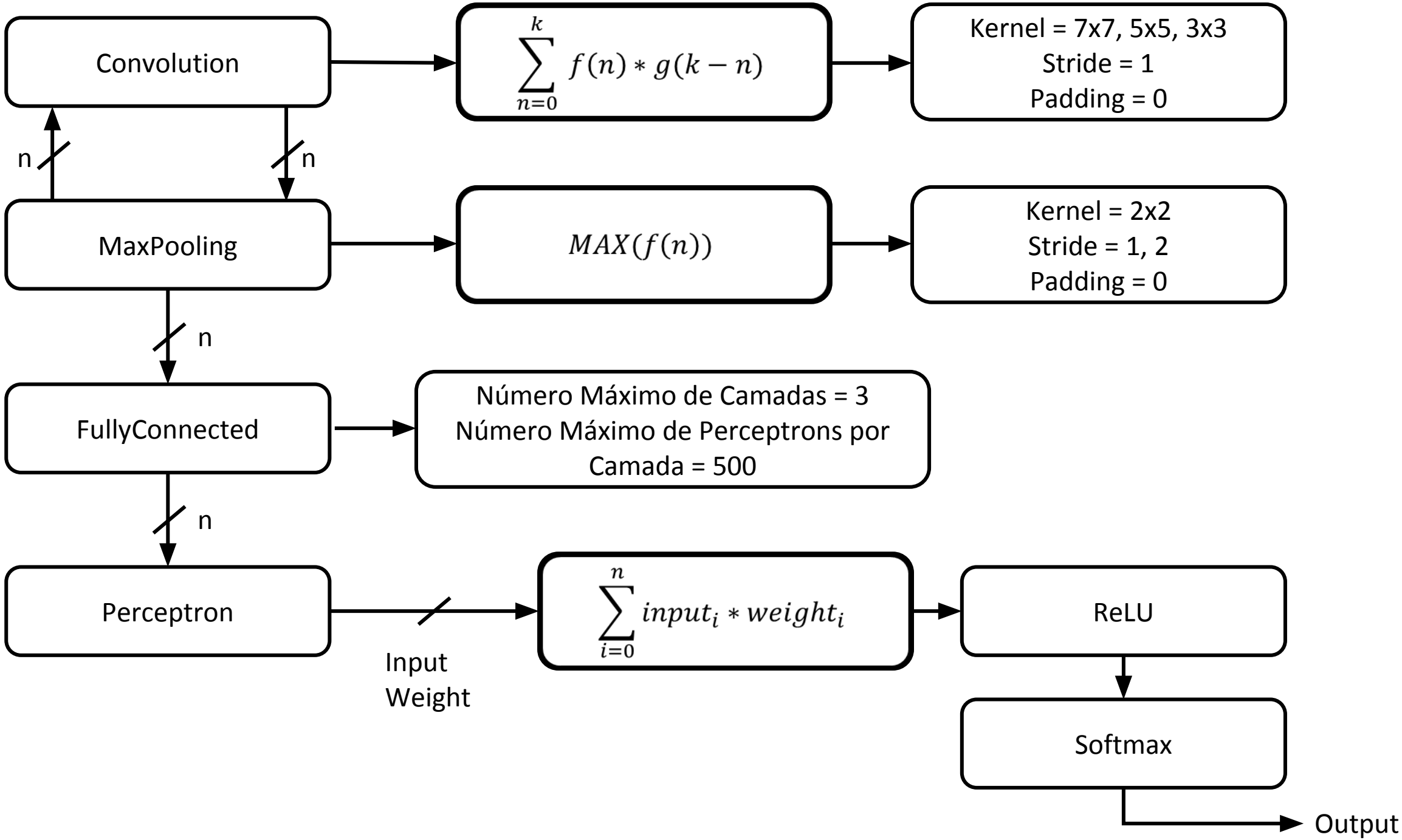

Convolution
n
n
MaxPooling
n
FullyConnect
Número Máximo de Camadas = 3
Número Máximo de Perceptrons por Camada = 500


Convolution
n
n
MaxPooling
n
FullyConnected
n
Perceptron


Convolution
n
n
MaxPooling
n
FullyConnected
n
Perceptron
Input
Weight
$\sum_{i=0}^{n} input_i * weight_i$


Convolution
n
n
MaxPooling
n
FullyConnected
n
Perceptron
Input
Weight
$\sum_{i=0}^{n} input_i * weight_i$
ReLU


Convolution
n
n
MaxPooling
n
FullyConnected
n
Perceptron
Input
Weight
$\sum_{i=0}^{n} input_i * weight_i$
ReLU
Softmax
Output


Convolution
$\sum_{n=0}^{k} f(n) * g(k-n)$
Kernel = 7x7, 5x5, 3x3
Stride = 1
Padding = 0
n
n
MaxPooling
$MAX(f(n))$
Kernel = 2x2
Stride = 1, 2
Padding = 0
n
FullyConnected
Número Máximo de Camadas = 3
Número Máximo de Perceptrons por Camada = 500
n
Perceptron
Input
Weight
$\sum_{i=0}^{n} input_i * weight_i$
ReLU
Softmax
Output

Sinais

- Input -> Imagem 32x32 em gray scale (CIFAR10);
- Weights -> Matriz definida pelo usuário;
- Quantidade de camadas convolucionais -> máx 3;
- Quantidade de camadas de pooling -> máx 3;
- Quantidade de camadas da fully connected;
- Quantidade de neurônios em cada camada;
- Tamanho do kernell das convoluções;
- Stride do máx pooling;
- Output.

O que foi feito ?

- Perceptron Parametrizado;
- MLP Não Parametrizada;
- Ativação com ReLU.

O que falta ?

- MLP Parametrizada;
- Camadas de Pooling e Convolução;
- Softmax.

Dificuldades

- Liberdade de projeto;
- Parametrização;
- Ponto flutuante (quantização).