

**МАШИННОЕ ОБУЧЕНИЕ  
И АНАЛИЗ ДАННЫХ**  
**(Machine Learning and Data Mining)**

Н. Ю. Золотых

<http://www.uic.unn.ru/~zny/ml>



*Лекция 11*

**Глубокое обучение**

## 11.1. Определения

(Yann LeCun, Yoshua Bengio, Geoffrey Hinton и др.)

*Глубокое обучение* (Deep learning) — подход, основанный на моделировании высокоуровневых абстракций (новых признаков) с помощью последовательных нелинейных преобразований.

Более высокие уровни нейронной сети представляют абстракцию на базе предыдущих слоев.

### **Некоторые подходы в глубоком обучении**

- Сверточные нейронные сети
- Автокодировщики (autoencoders) и стеки автокодировщиков
- Ограниченная машина Больцмана и глубокие сети доверия (deep belief networks)

### **Проблемы с большими нейронными сетями**

- Переобучение
- Исчезающий градиент

## 11.1.1. Сверточные нейронные сети





Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Основная идея сверточных нейронных сетей (сверточных слоев):

Параметры фильтра будем подбирать с помощью обучения:

$$z_{pq} = \sigma \left( \beta_0 + \sum_{i=1}^h \sum_{j=1}^w \beta_{ij} x_{p+i-1, q+j-1} \right)$$

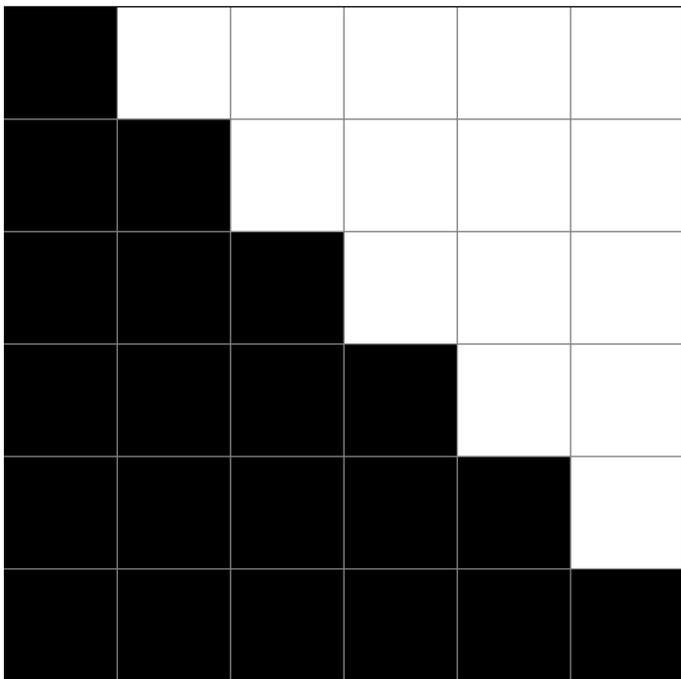
$x_{ij}$  — узлы (нейроны) одного слоя (например, входного)

$z_{pq}$  — узлы следующего слоя

Параметры фильтра — это теперь веса нейронной сети.

Отличия от полносвязной сети (полносвязного слоя):

- Нет соединения каждого узла одного слоя со всеми узлами следующего.
- Веса становятся *разделяемыми*.



0	1	0
1	-4	1
0	1	0

Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

0	255	255	255	255	255
0	0	255	255	255	255
0	0	0	255	255	255
0	0	0	0	255	255
0	0	0	0	0	255
0	0	0	0	0	0

0	1	0
1	-4	1
0	1	0

Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

0	255	255	255	255	255
0	0	255	255	255	255
0	0	0	255	255	255
0	0	0	0	255	255
0	0	0	0	0	255
0	0	0	0	0	0

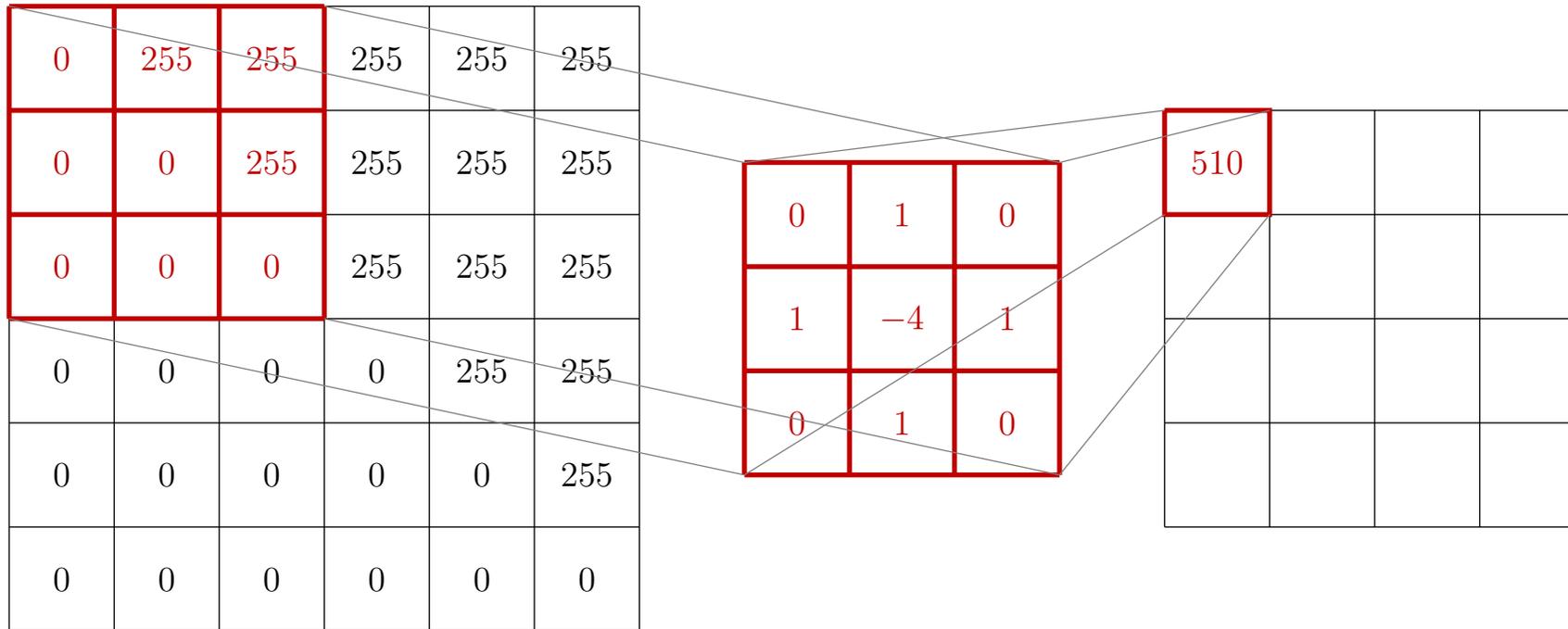
0	1	0
1	-4	1
0	1	0


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

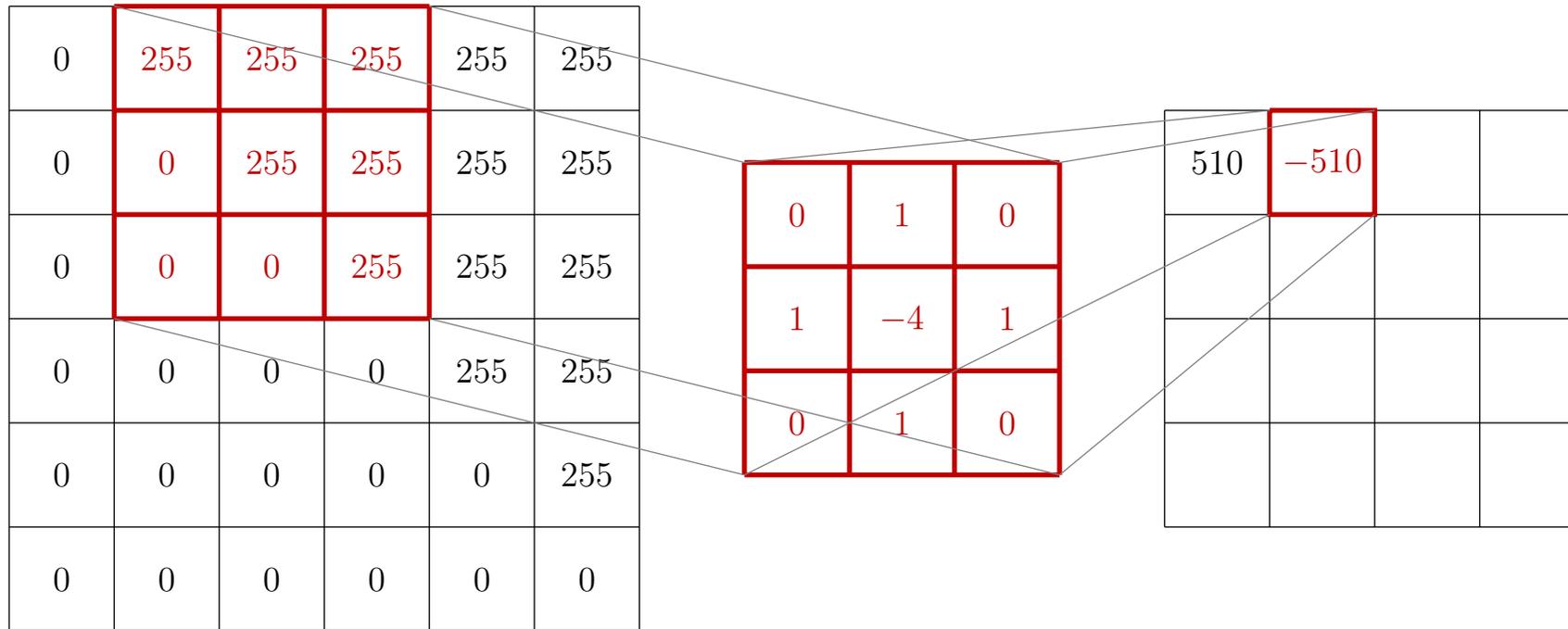


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

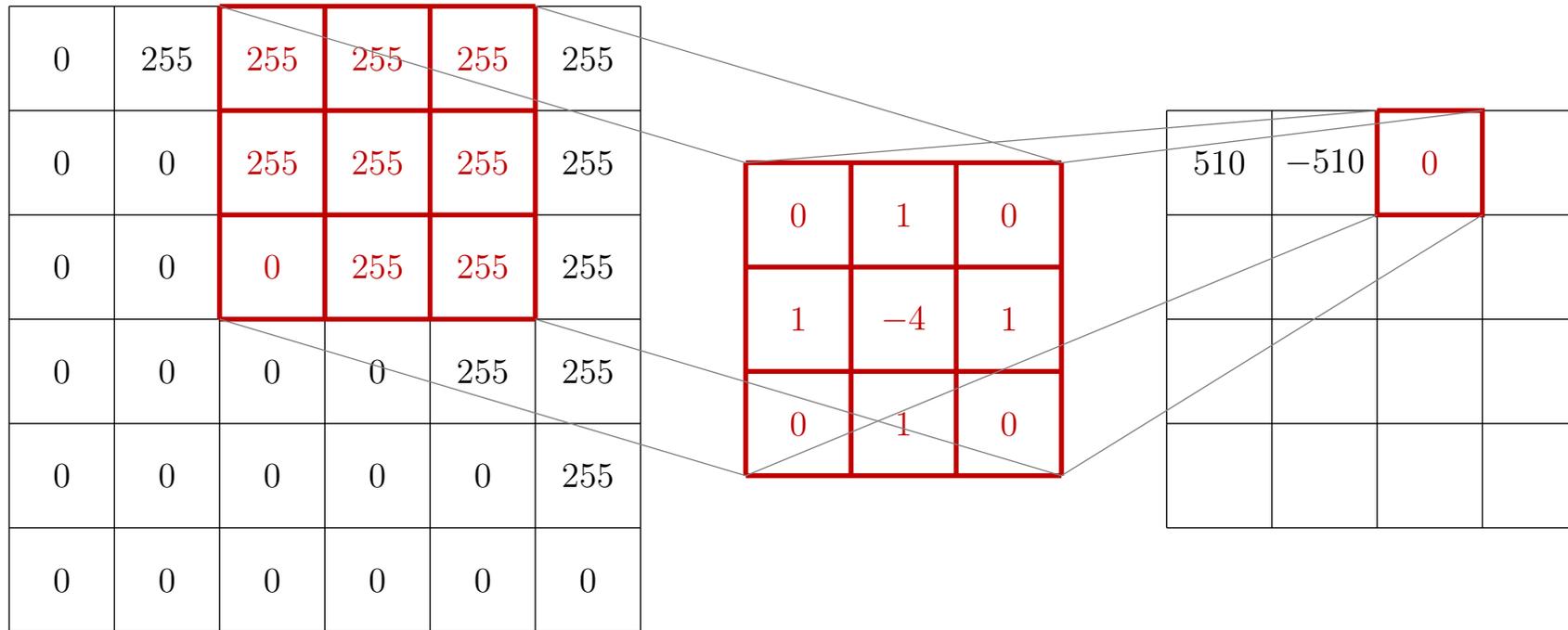


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

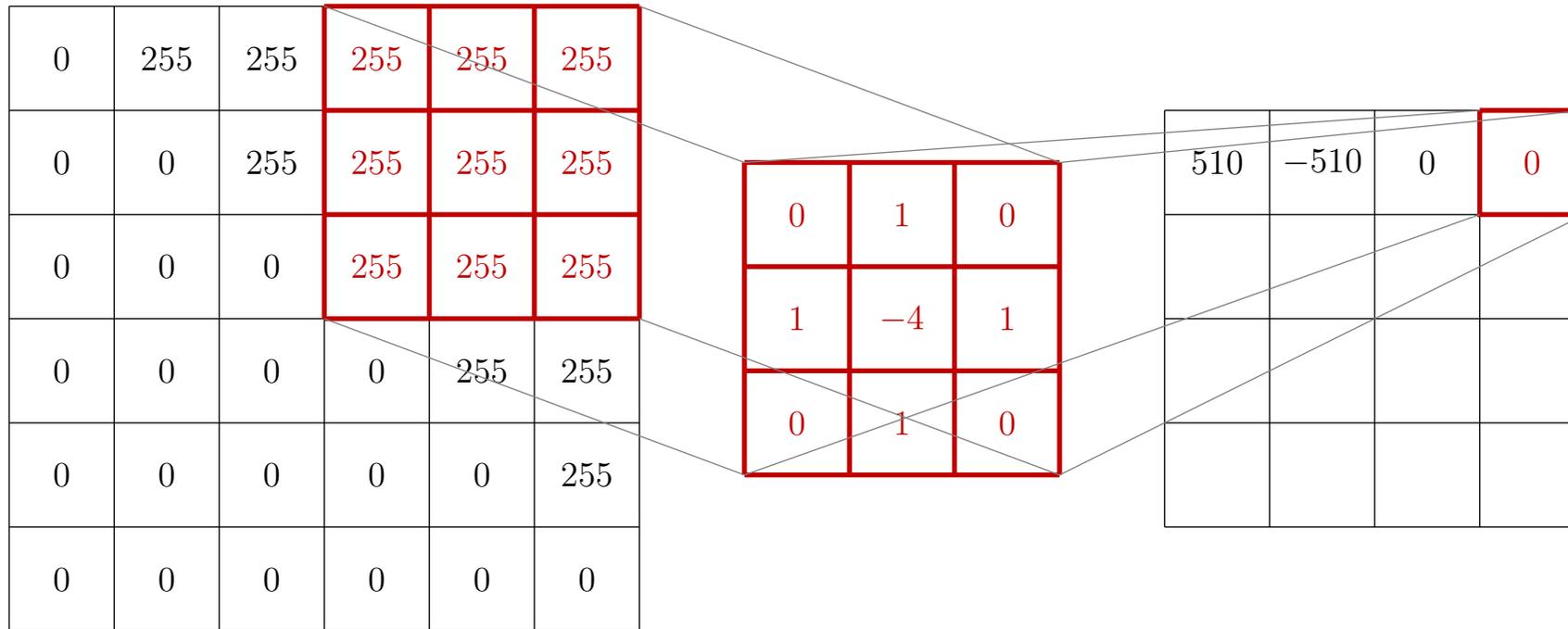


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

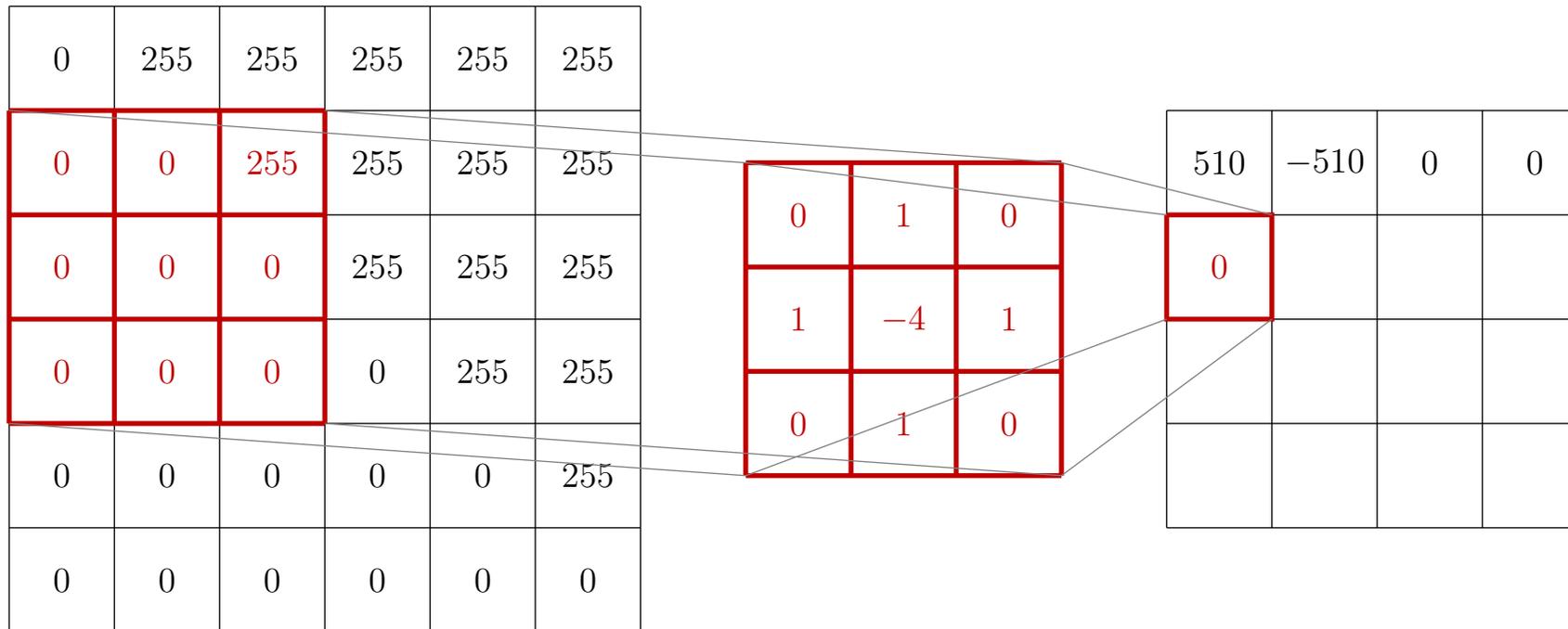


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

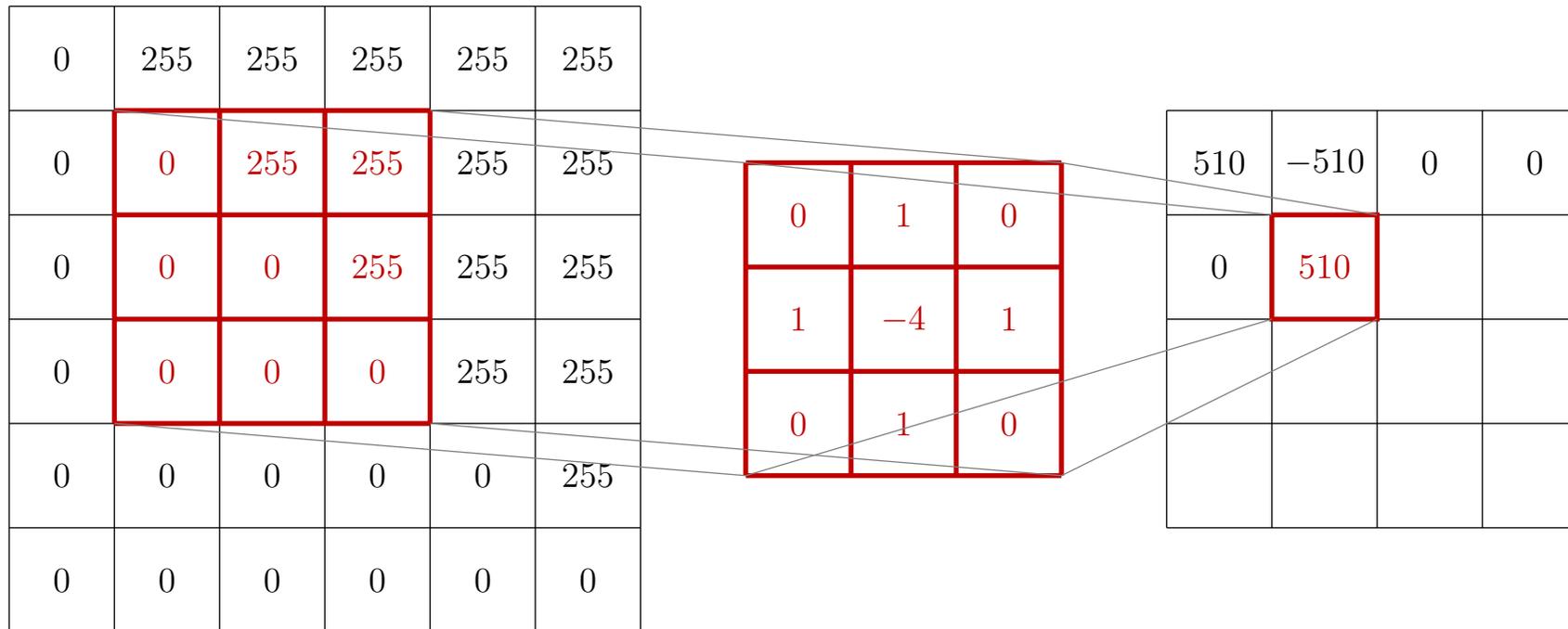


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

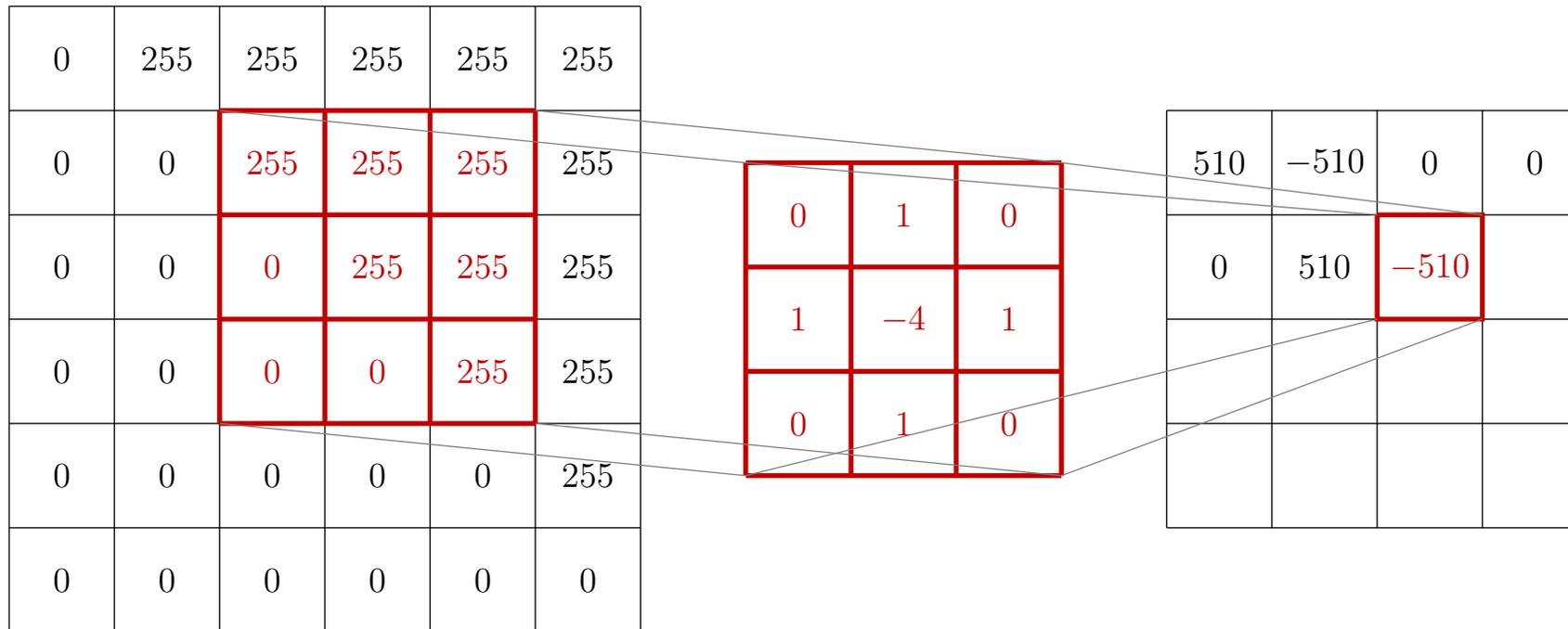


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

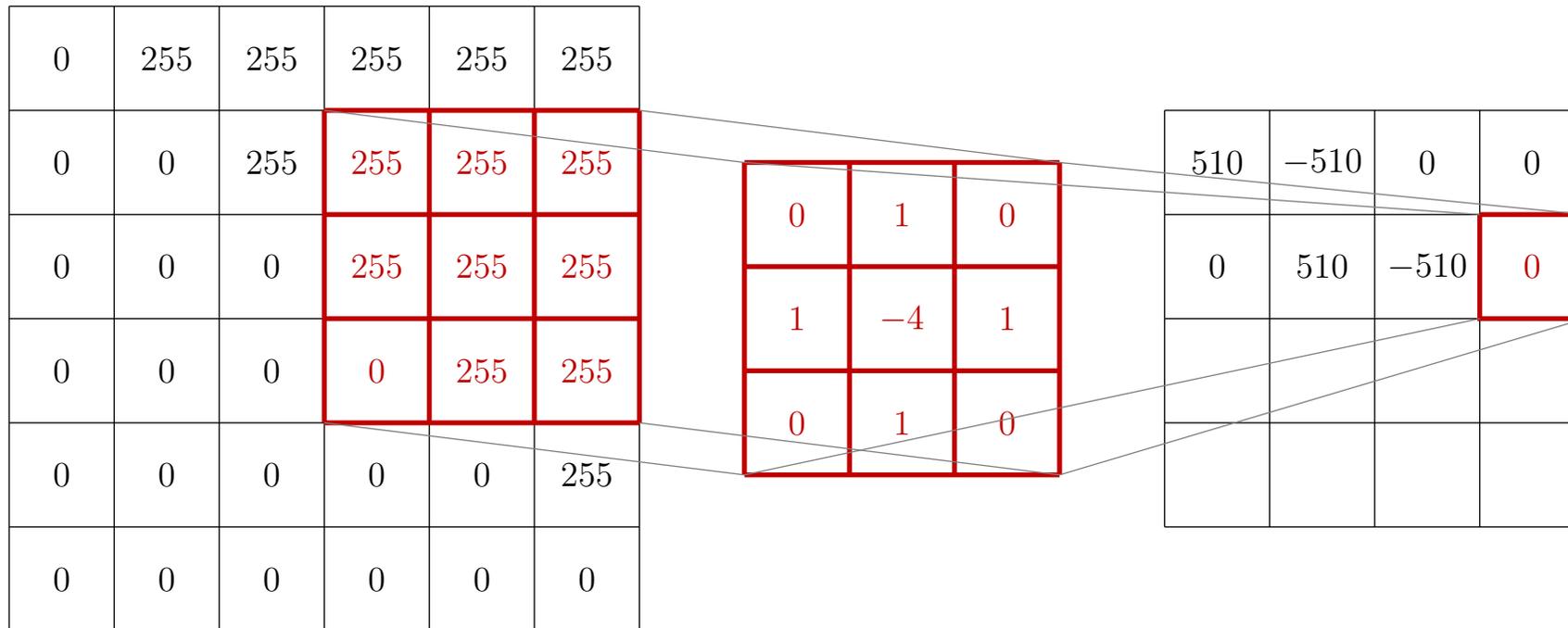


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

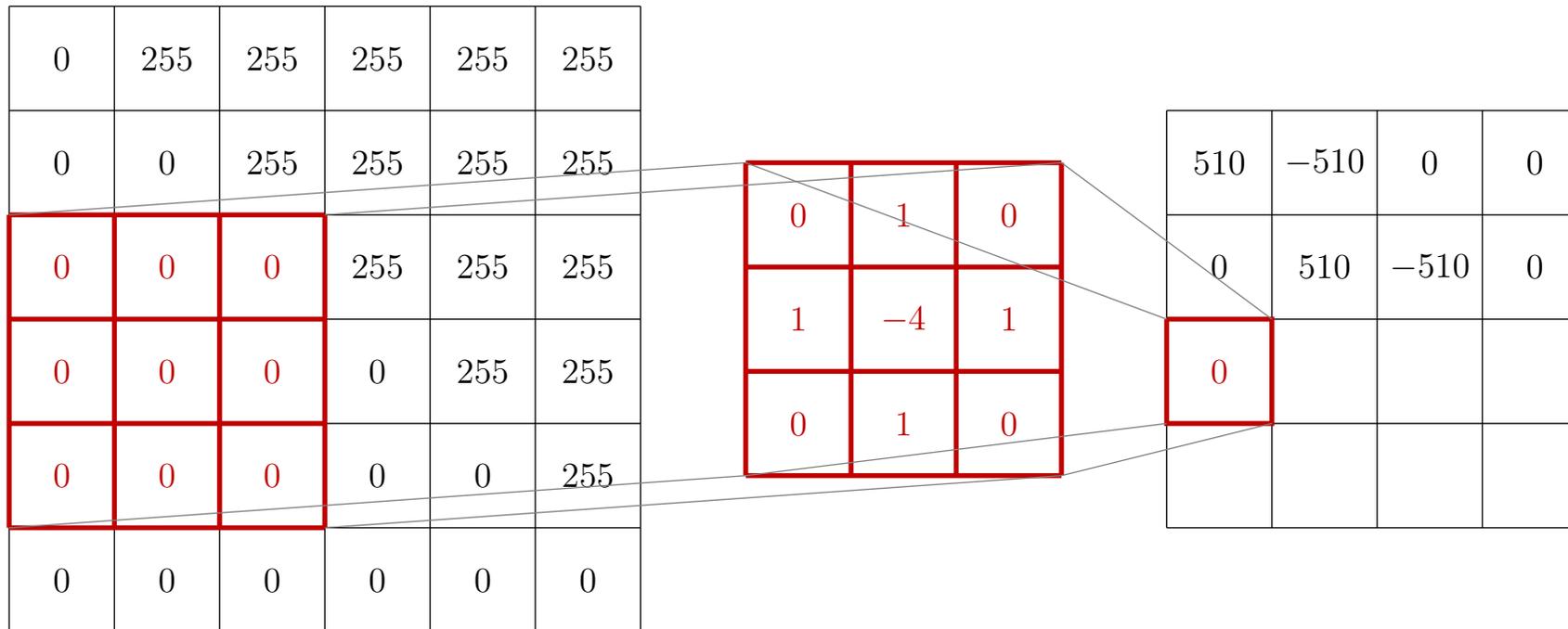


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

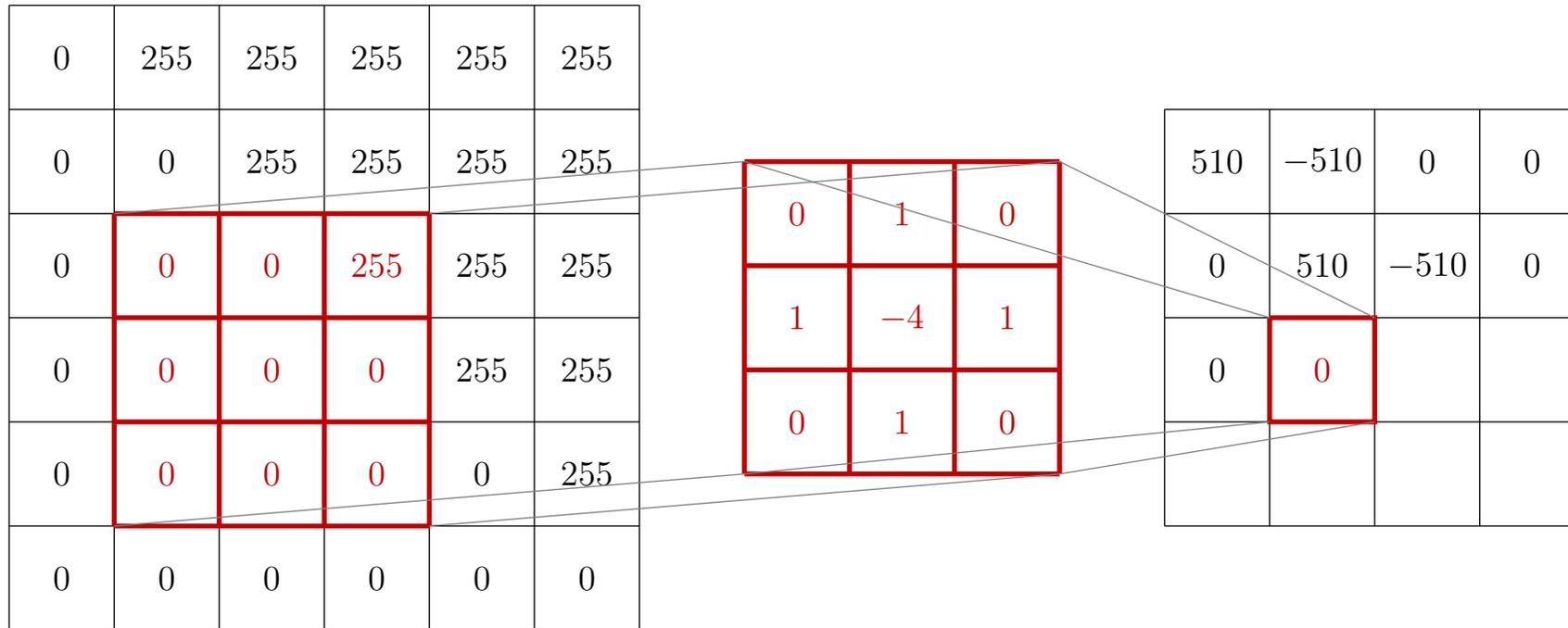


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

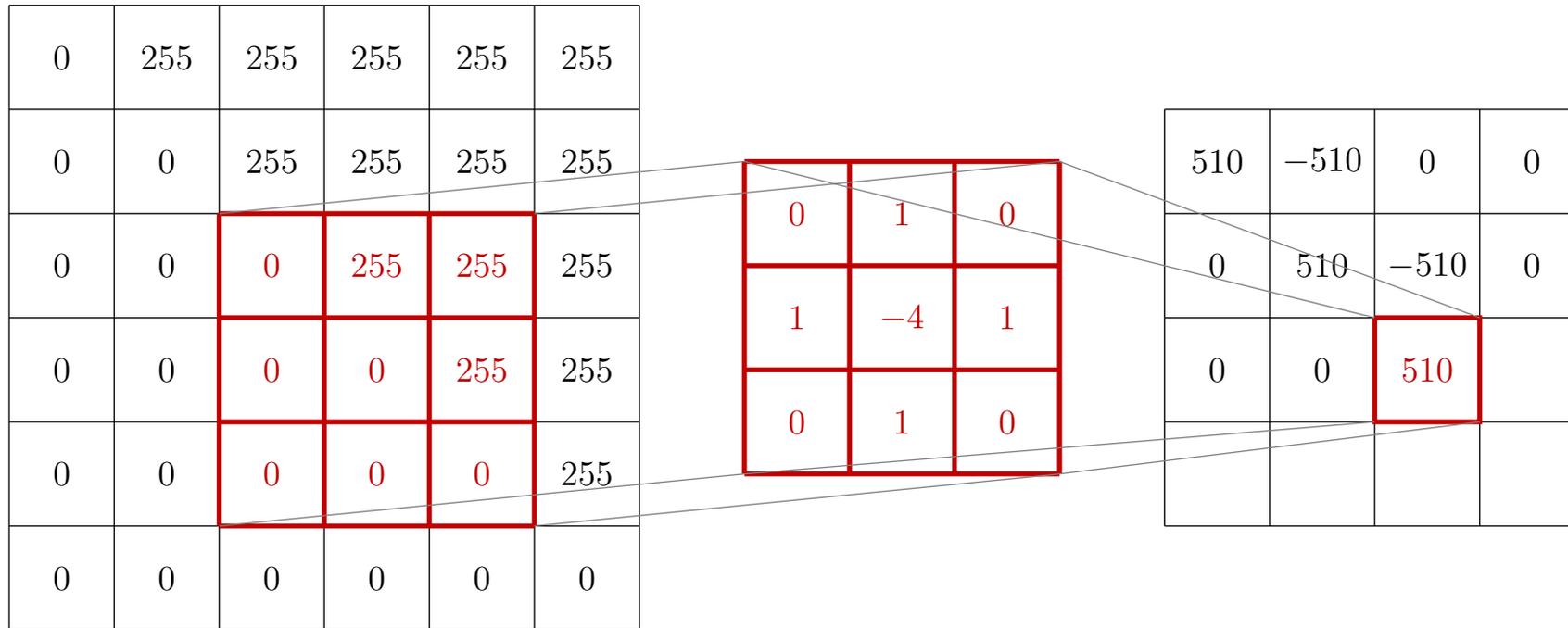


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

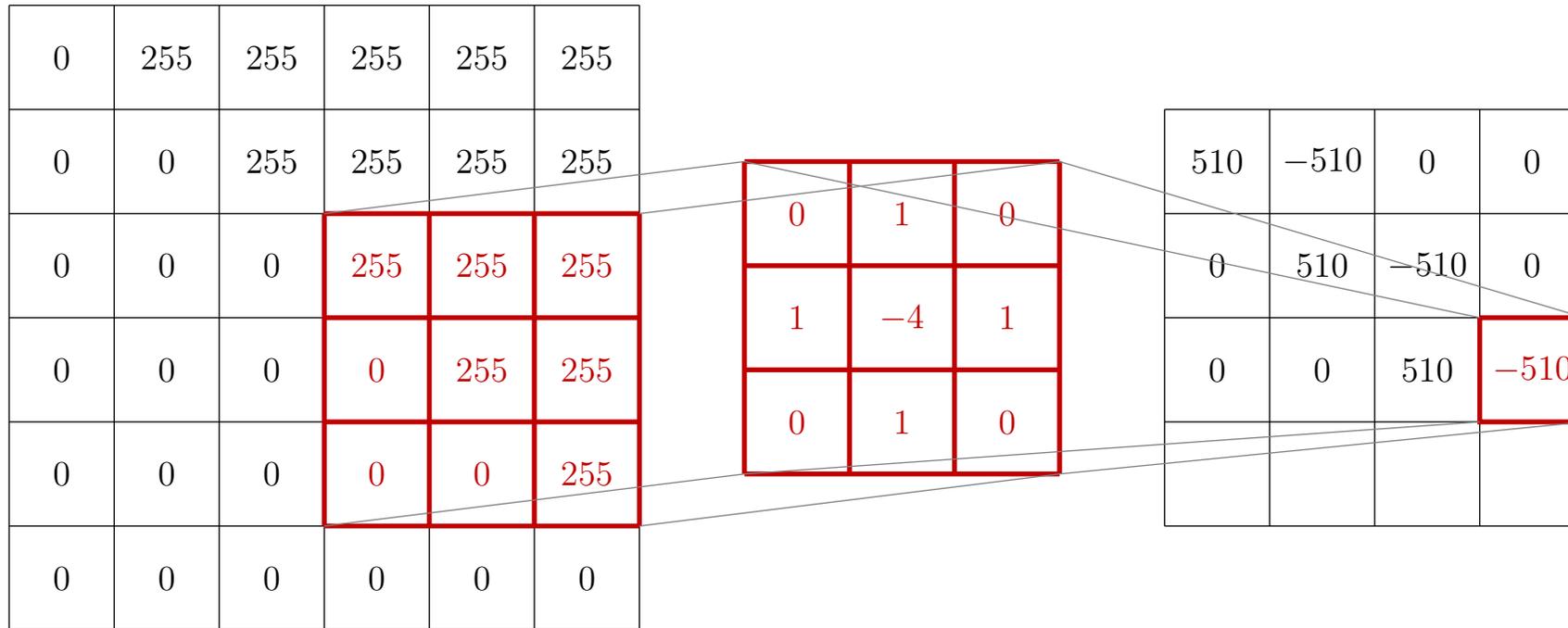


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

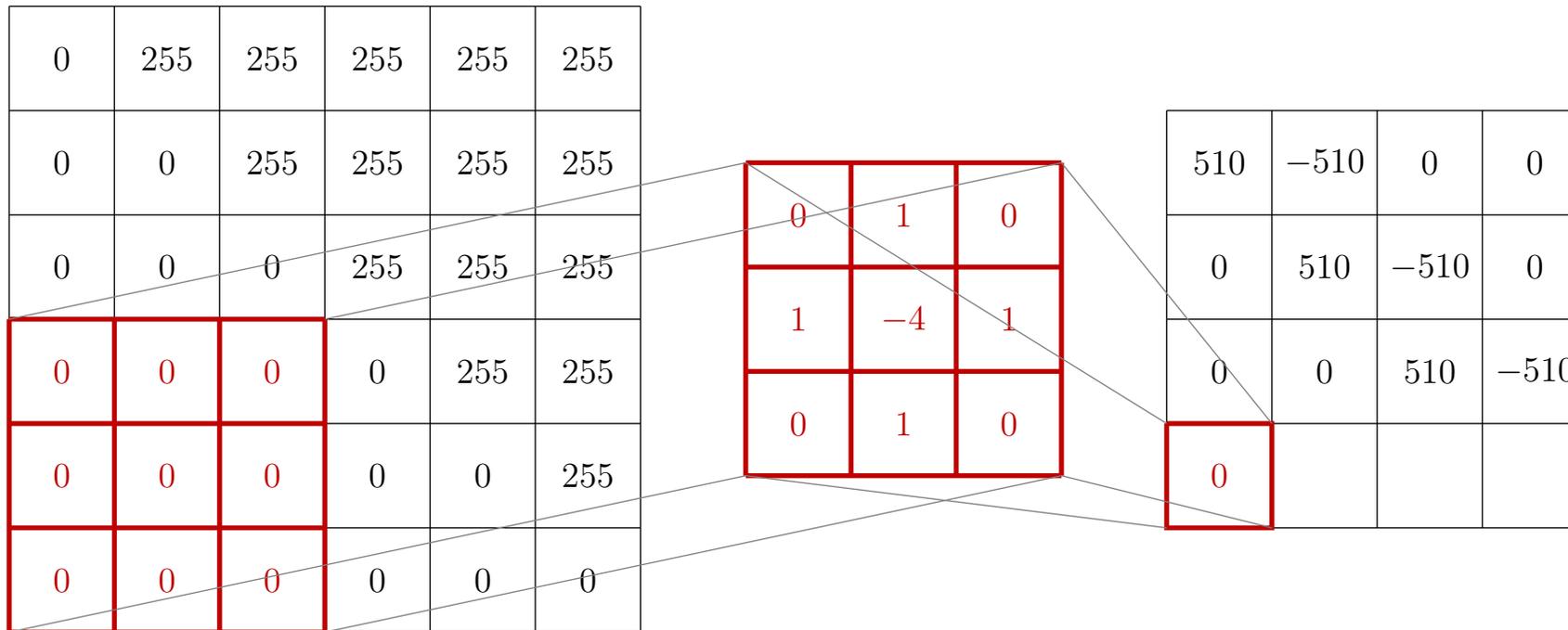


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

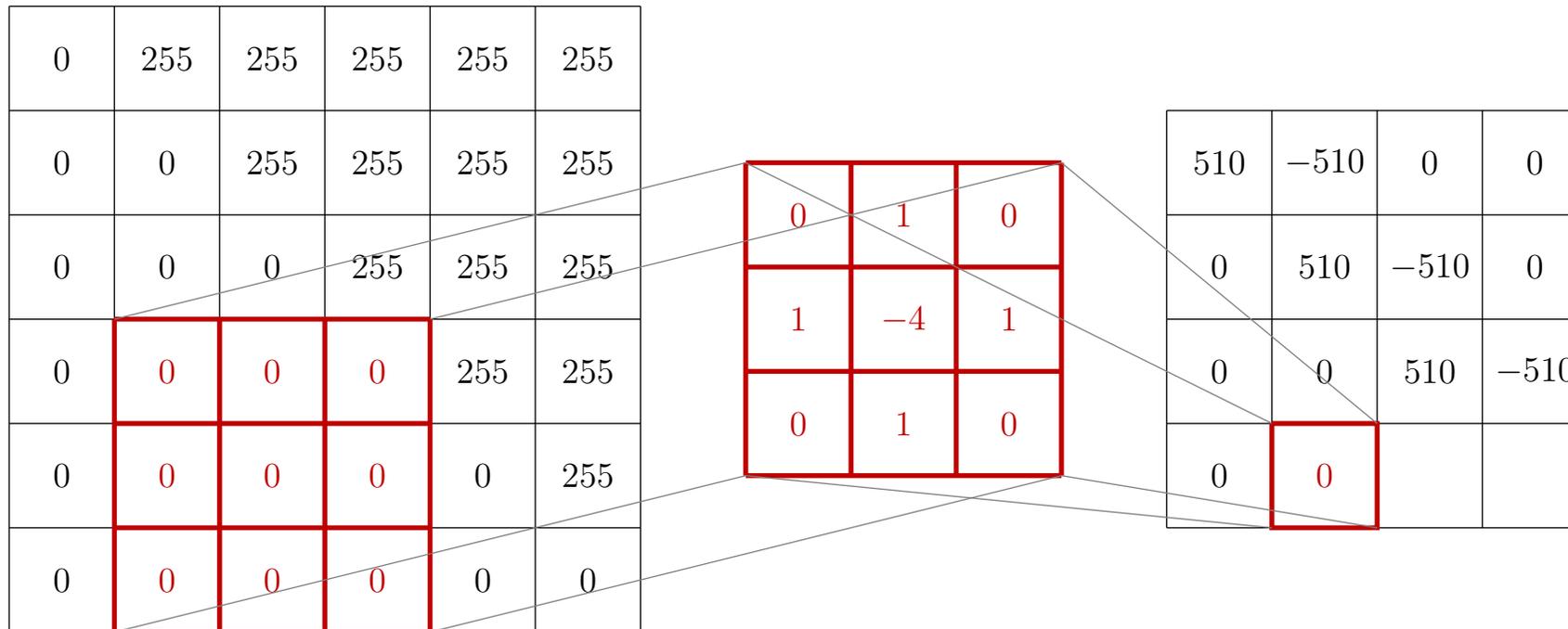


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

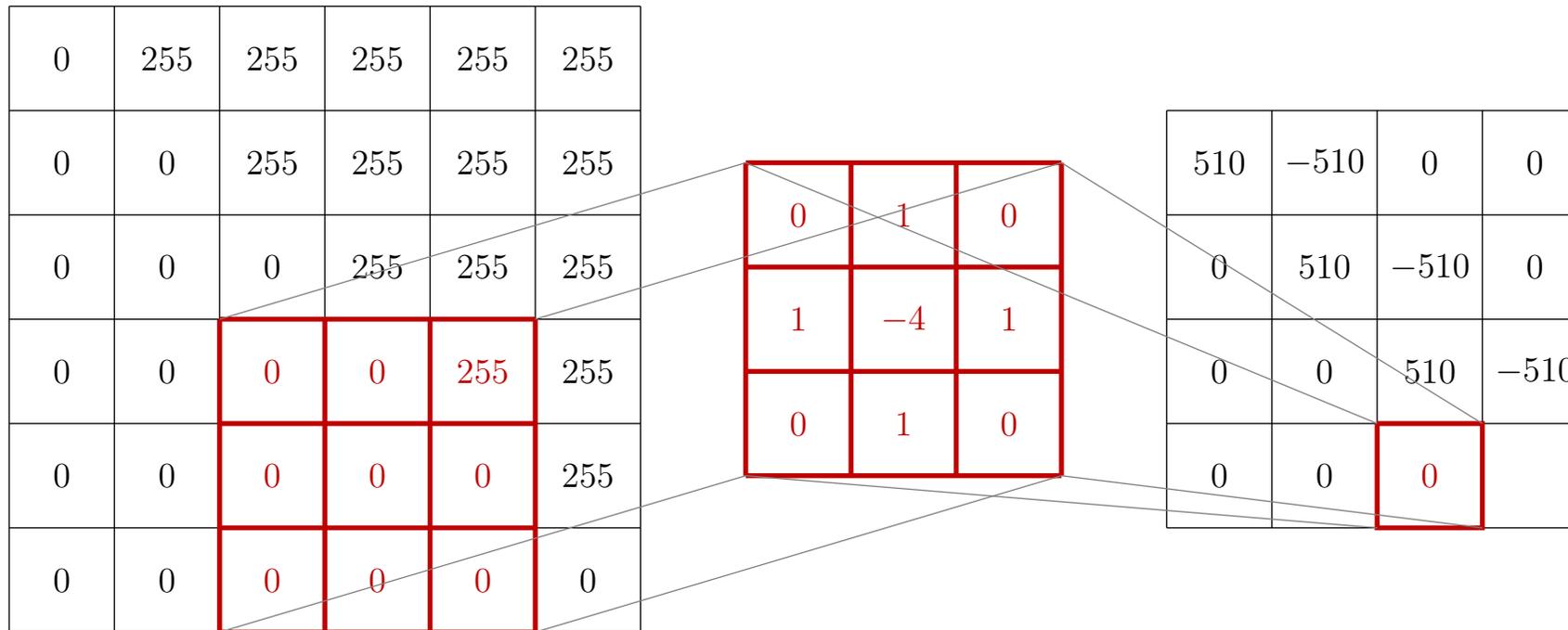


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

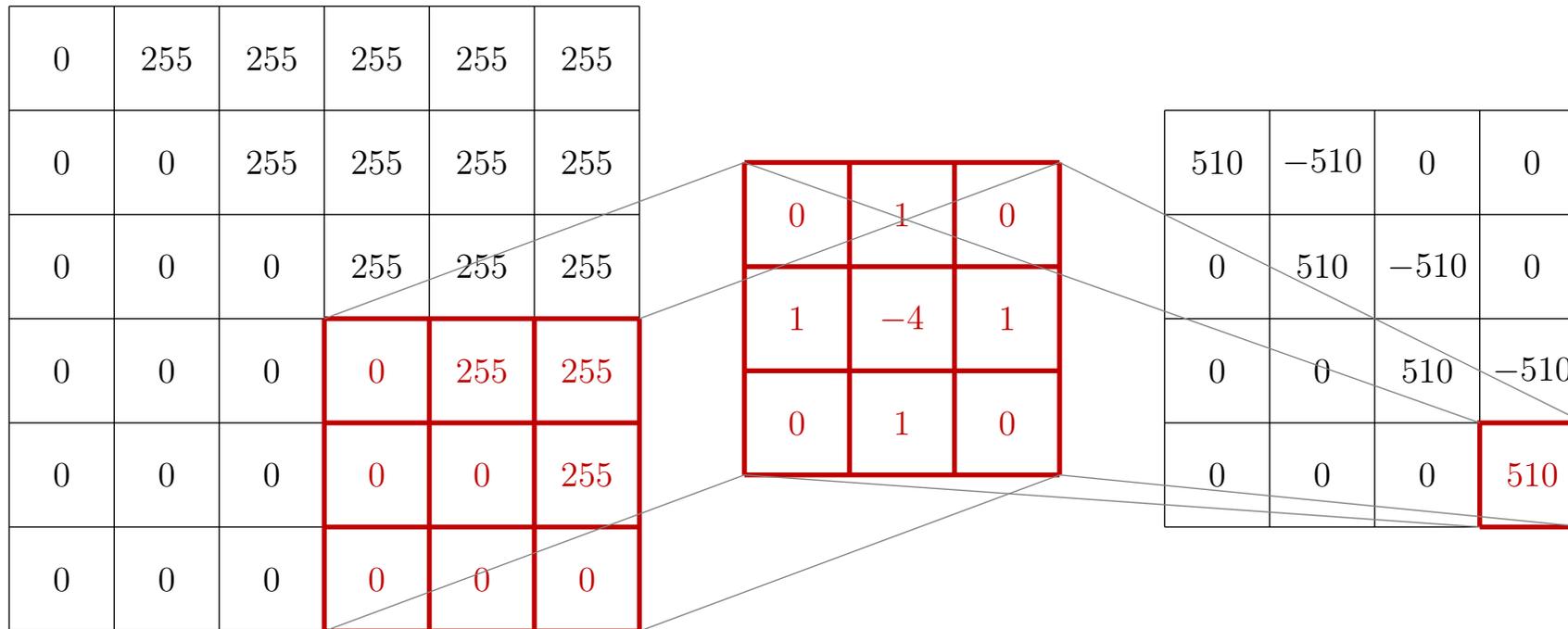


Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

0	255	255	255	255	255
0	0	255	255	255	255
0	0	0	255	255	255
0	0	0	0	255	255
0	0	0	0	0	255
0	0	0	0	0	0

0	1	0
1	-4	1
0	1	0

510	-510	0	0
0	510	-510	0
0	0	510	-510
0	0	0	510

Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

0	255	255	255	255	255
0	0	255	255	255	255
0	0	0	255	255	255
0	0	0	0	255	255
0	0	0	0	0	255
0	0	0	0	0	0

0	1	0
1	-4	1
0	1	0

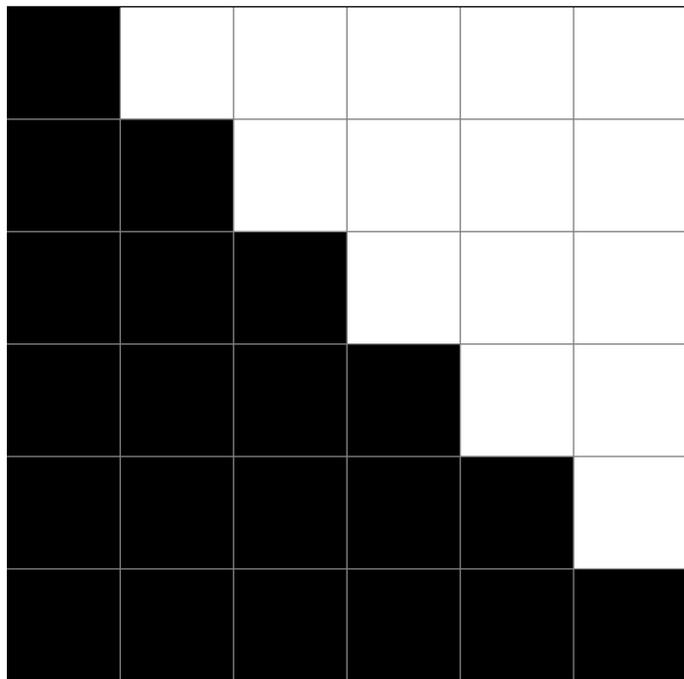
255	0	0	0
0	255	0	0
0	0	255	0
0	0	0	255

Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

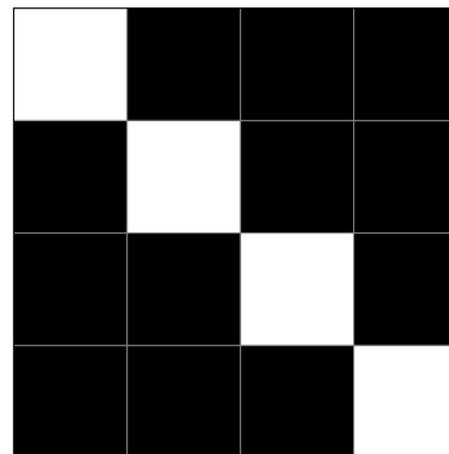
$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



0	1	0
1	-4	1
0	1	0



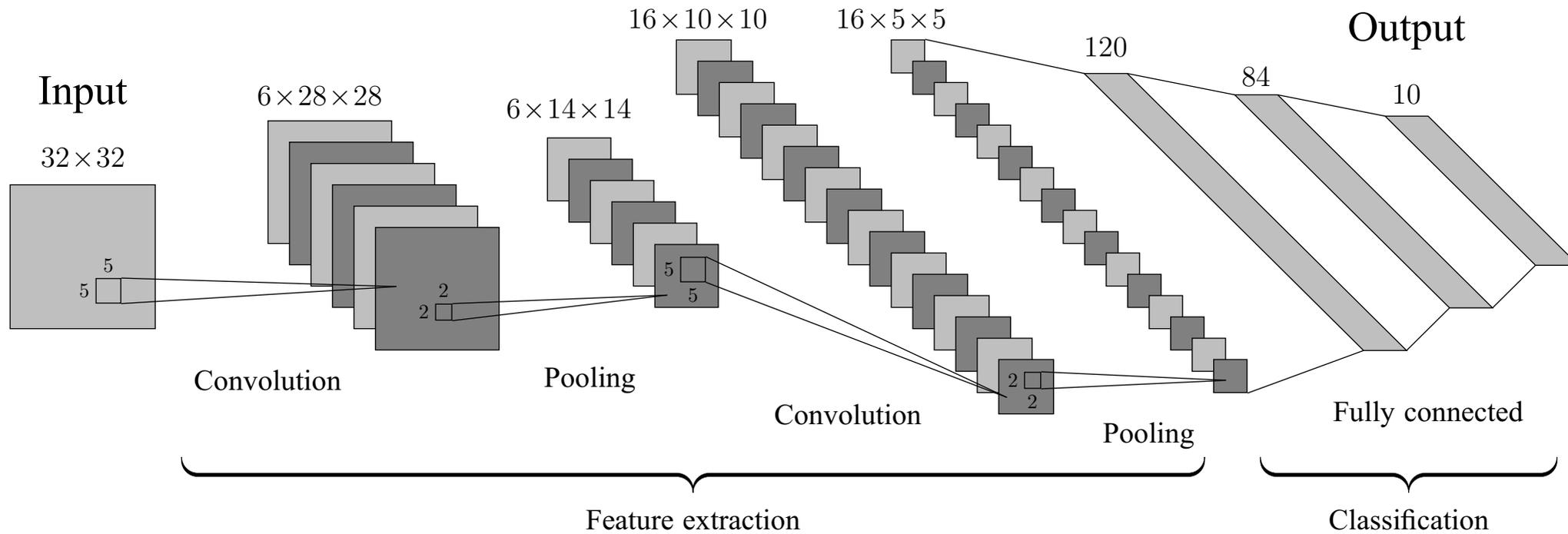
Линейный фильтр (свертка)  $I * K$  с ядром  $K$ :

$$(I * K)_{pq} = \sum_{i=1}^h \sum_{j=1}^w I_{p+i-1, q+j-1} K_{ij}$$

Например,

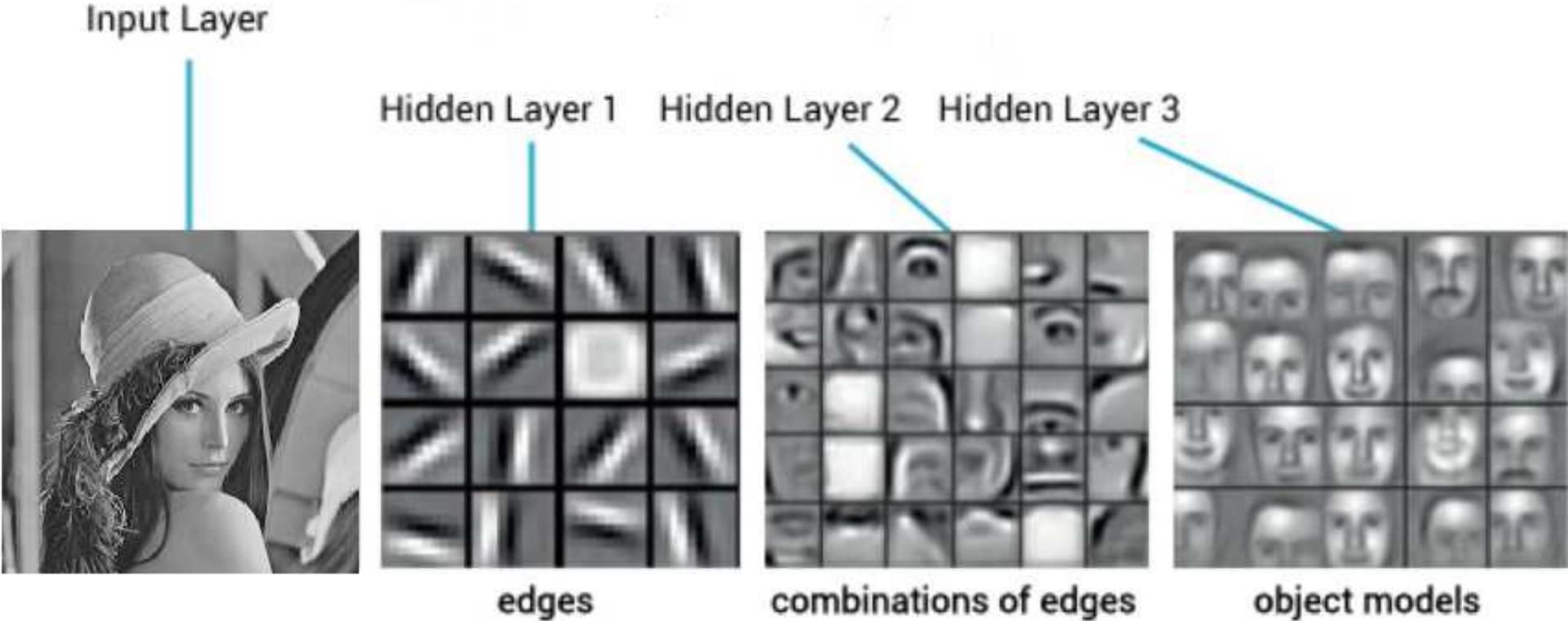
$$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

# LeNet-5 [Le Cun et al., 1998]



- *Сверточные слои (convolutional layers)*
- *«Выборочные» слои, или слои объединения (subsampling/pooling layers)*
- *Полносвязные слои (fully connected layers)*

Выделение признаков в слоях нейронной сети (схема)



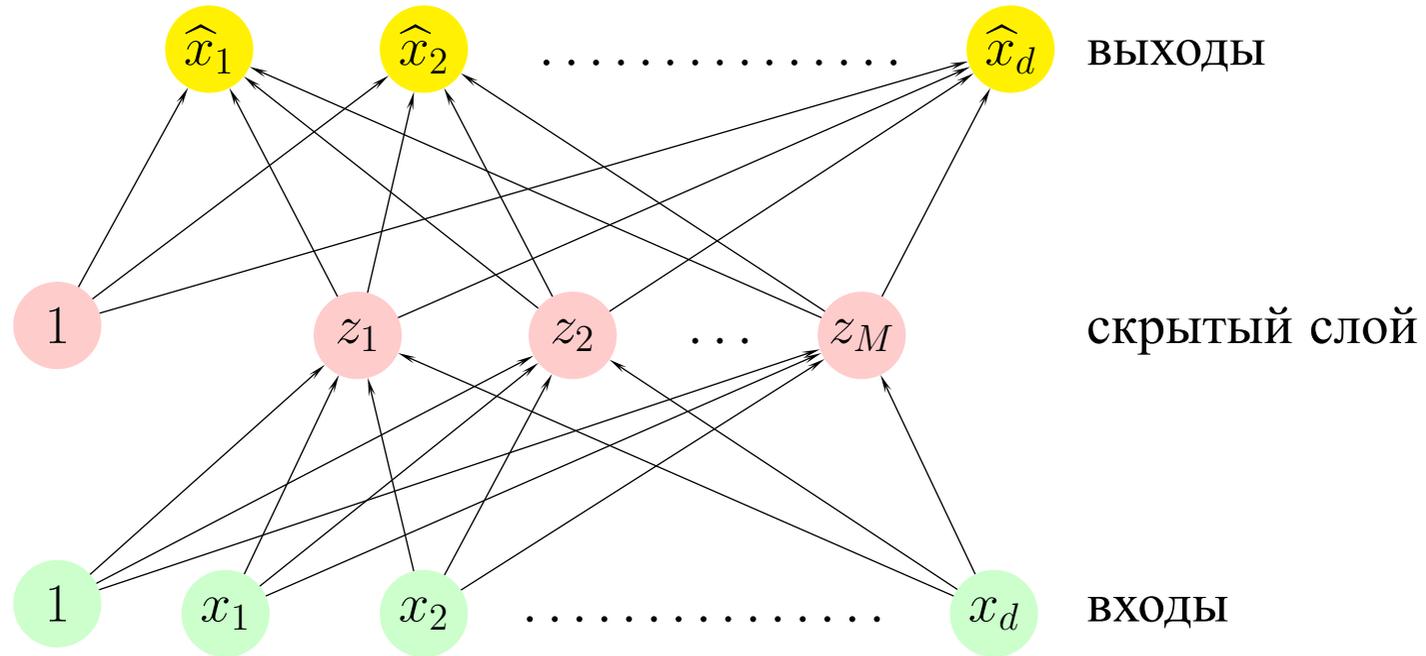
Другие примеры:

## ImageNet Large-Scale Visual Recognition Challenge

- AlexNet (Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton, 2012) — победитель ImageNet-2012 — 8 слоев, 61 млн. параметров
- GoogLeNet (Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, 2015) — 22 слоя, 7 млн. параметров
- Microsoft ResNet (Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015) — 101 слой, 44.6 млн. параметров
- DenseNet (Gao Huang, Zhuang Liu, Laurens Van Der Maaten, Kilian Q. Weinberger, 2017) — 201 слой, 20 млн. параметров
- ...
- Playing Atari with Deep Reinforcement Learning (V. Mnih, 2013) — 10 слоев
- AlphaGo (D.Silver et al, Alphabet Inc.'s Google DeepMind, 2015) — 2 нейронных сети по 13 слоев

## 11.1.2. Автокодировщики

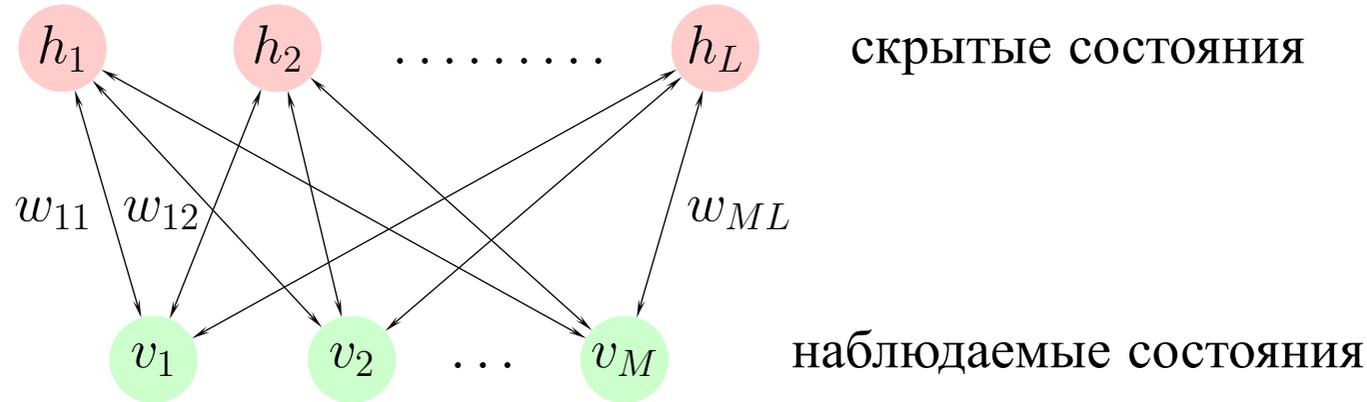
Автокодировщик (*autoencoder*) — метод сжатия (с потерями) на основе обучения без учителя.



$$x_1 \approx \hat{x}_1, \quad x_2 \approx \hat{x}_2, \quad \dots, \quad x_d \approx \hat{x}_d$$

$M < d$  или используется разреженность

### 11.1.3. Ограниченная машина Больцмана



*Метод сравнительного отклонения (Contrastive Divergence)*

**begin**

Наблюдаемым состояниям присваиваются значения  $v = (v_1, v_2, \dots, v_M)$

По ним вычисляются значения скрытого слоя  $h = (h_1, h_2, \dots, h_L)$

По  $h$  вычисляются новые значения наблюдаемого слоя  $v' = (v'_1, v'_2, \dots, v'_M)$

По  $v'$  вычисляются новые значения скрытого слоя  $h' = (h'_1, h'_2, \dots, h'_L)$

$W \leftarrow W + \alpha(vh^\top - v'h'^\top)$

**end**

$$v_j, h_i \in \{0, 1\}, \quad \Pr \{h_i = 0\} = \sigma \left( \sum_{j=1}^M w_{ij} v_j \right), \quad \Pr \{v'_i = 0\} = \sigma \left( \sum_{i=1}^L w_{ij} h_i \right)$$

## 11.2. Программное обеспечение

1. TensorFlow
2. Keras
3. PyTorch
4. Caffe
5. ...