Convolutional neural networks

Lecture 7

Changho Suh

January 24, 2024

Convolutional neural networks (CNNs) and history

Recap: Data organization & evaluation

m is a deciding factor for the data split **ratio**:

4 regimes: Small, middle, large and ultra-large

val set dist. ~ test set dist. ~ target dist.

Cross validation for evaluation:

Choose a hyperparameter that minimizes the average validation loss:

$$\text{val loss} = \frac{\text{val}_1 + \text{val}_2 + \text{val}_3 + \text{val}_4}{4}$$

Recap: Techniques for DNNs

Weight initialization: He's

Training stability: Adam optimizer
learning rate decaying, BN

Hyperparameter search:

L of layers, # $n^{[\ell]}$ of hidden neurons, activation

Recap: DNNs

Model complexity is very high.

This often leads to the overfitting problem.

This motivates the use of generalization techniques:

Regularization; data augmentation;

early stopping; drop out

Recap: Tensorflow coding

Train-val-test data split:

```
from sklearn.model_selection import train_test_split

X_,X_test,y_,y_test = train_test_split(X,y,test_size=1/10,stratify=y)

X_train,X_val,y_train,y_val = train_test_split(X_,y_,test_size=1/9,stratify=y_)
```

Generalization techniques:

```
from tensorflow.keras.regularizers import l2

from tensorflow.keras.callbacks import EarlyStopping

from tensorflow.keras.layers import Dropout
```

Recap: Tensorflow coding

Weight initialization:

```
from tensorflow.keras.initializers import HeNormal
```

Batch normalization:

```
from tensorflow.keras.layers import BatchNormalization
```

Learning rate decaying:

```
from tensorflow.keras.callbacks import LearningRateScheduler
```

Cross validation:

```
from sklearn.model_selection import KFold
```

What is next?

One important question:

Can DNNs be specialized?

CNNs: Image data

RNNs: Text/audio data (language) and any sequential data

Outline of today's lectures

Focus on CNNs.

Specifically we will:

1. Investigate how CNNs were developed;
2. Study the two key building blocks:

   Conv layer

   Pooling layer
3. Discuss two popular CNN architectures.

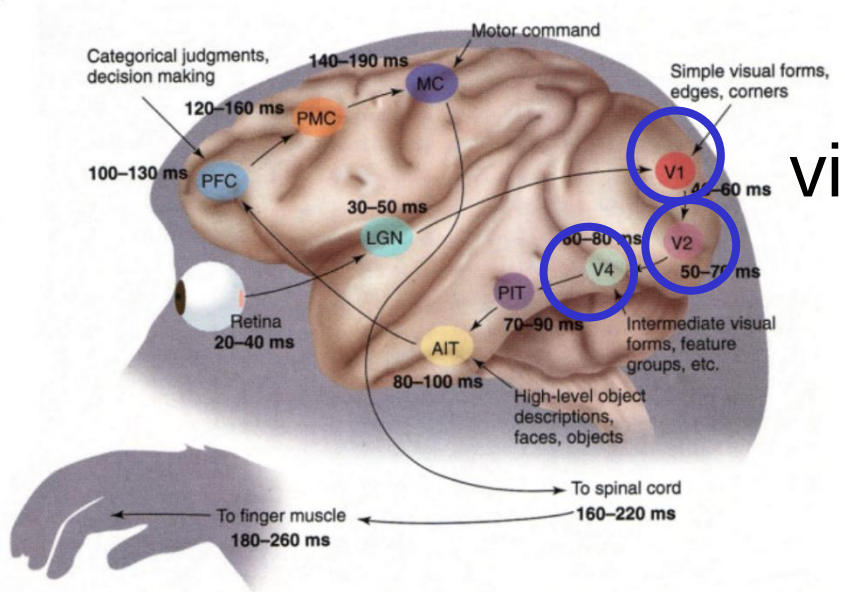
Focus of Lecture 7

Focus on CNNs.

Specifically we will:

1. Investigate how CNNs were developed;
2. Study the two key building blocks;
   - Conv layer
   - Pooling layer
3. Discuss two popular CNN architectures.

Visual cortex (시각피질)



visual cortex

Hubel & Wiesel ’58/’59 observed:

1. React only to a limited region (*receptive field)*
2. Exhibit hierarchy: V1→V2→V3→V4→ …

*Won them the **Nobel Prize** in Physiology or Medicine in **‘81**.

Led to the birth of CNNs

Developed the first CNN in 1980:
Neocognitron

Kunihiko Fukushima
1980

Developed another CNN in 1988:
LeNet-5

Commercialization: Handwritten check numbers recognition

Yann LeCun
1988

Two building blocks of CNNs

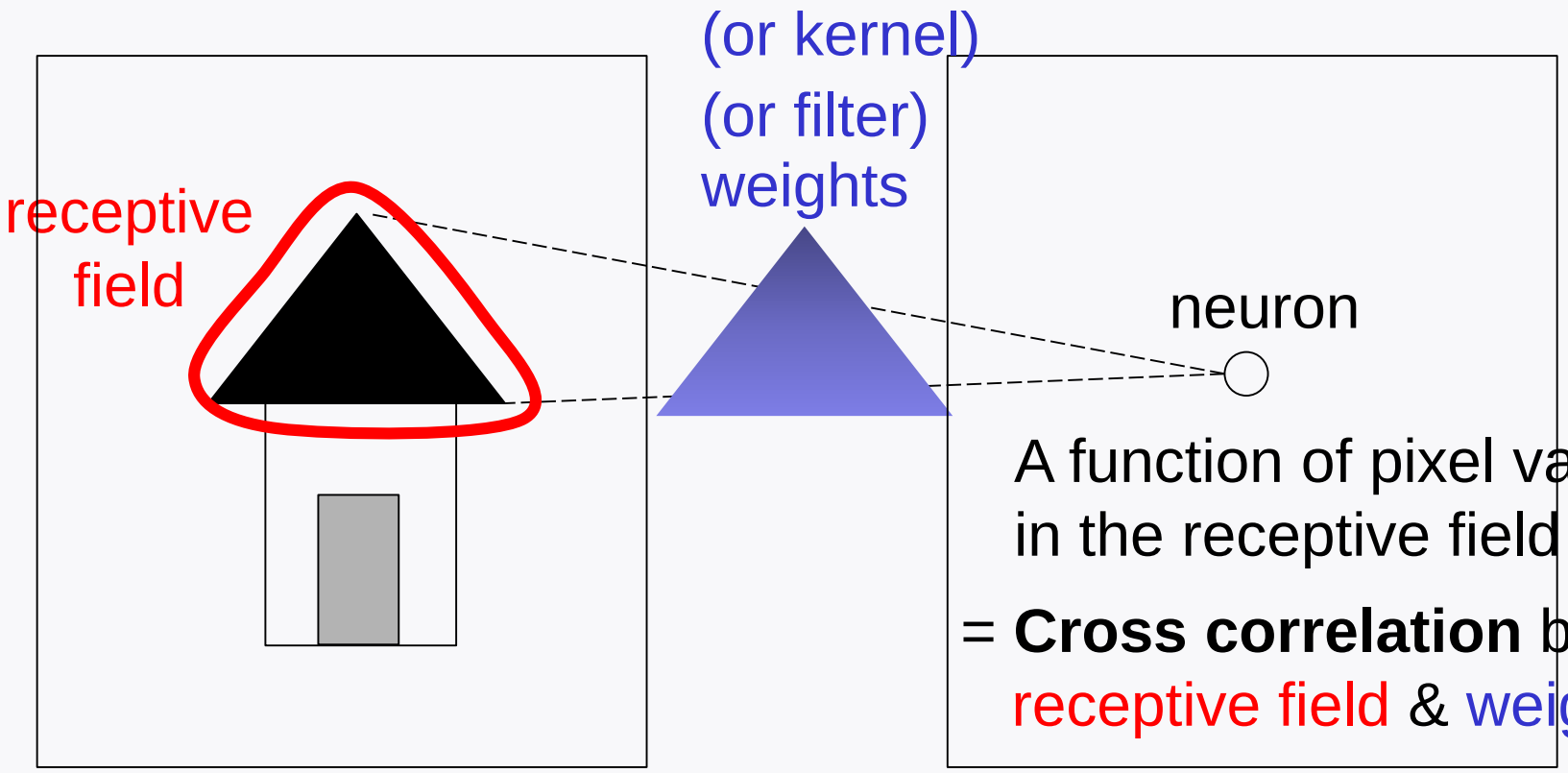
1. Convolutional layer (Conv layer)

Role: Mimick neurons' behaviors:
Reacting only to receptive fields.

2. Pooling layer

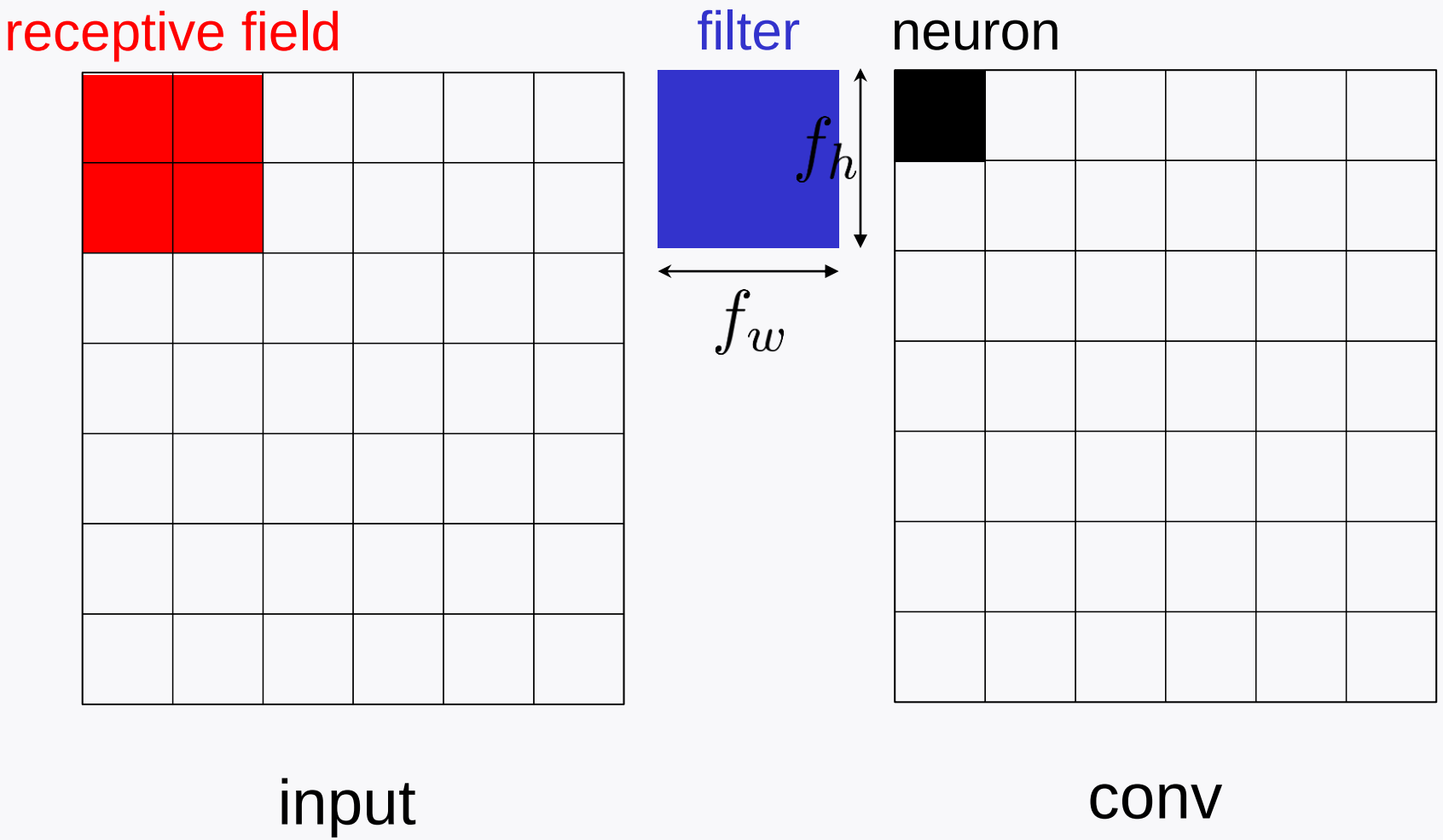
Role: Downsample to reduce complexity
(# parameters & memory size).

Conv layer

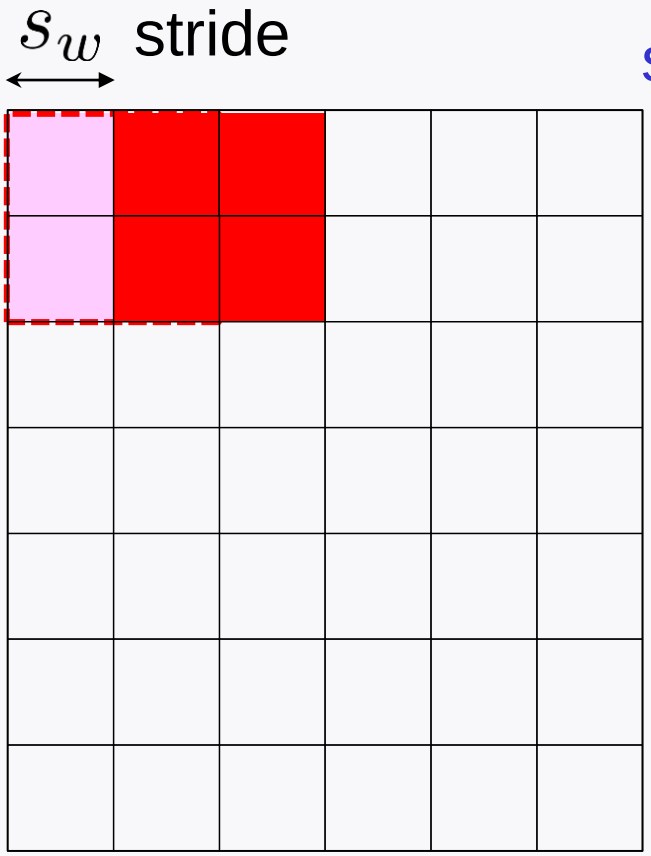


*Convolution operation is very similar to cross correlation.

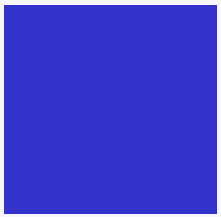
Abstraction via grids



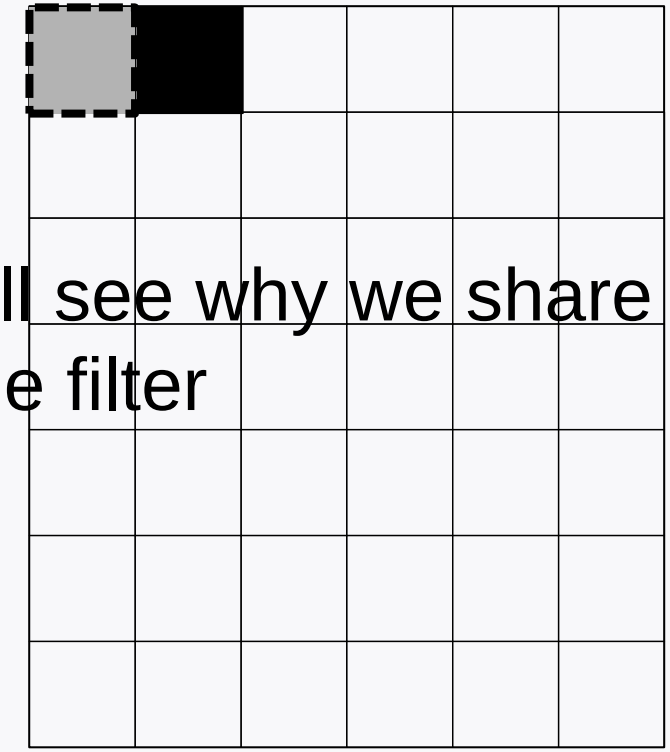
What about for next *right* neuron?



same filter



neuron

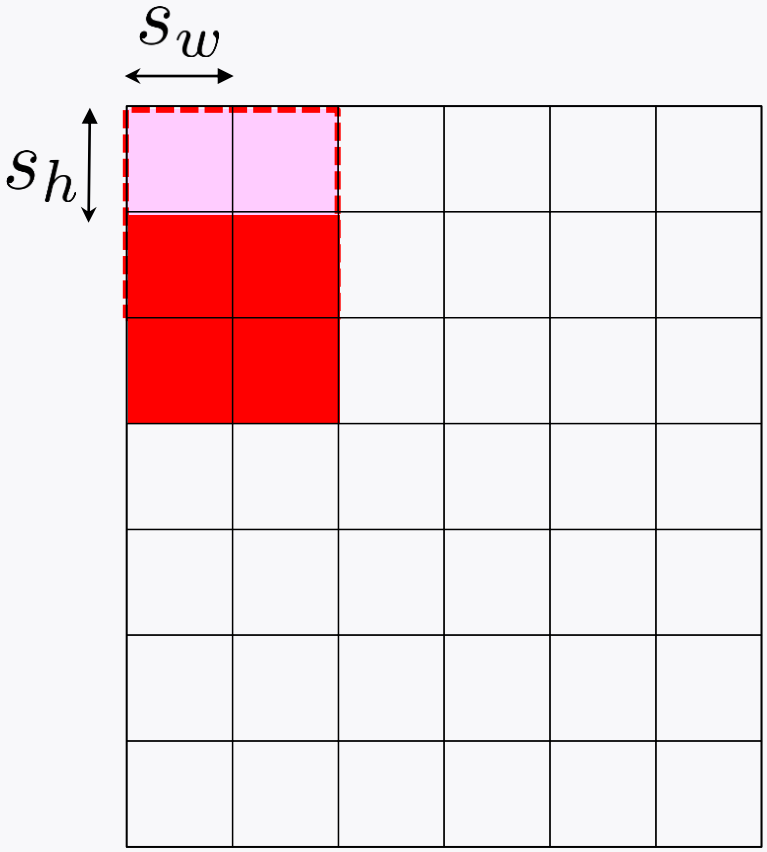


Later will see why we share the same filter

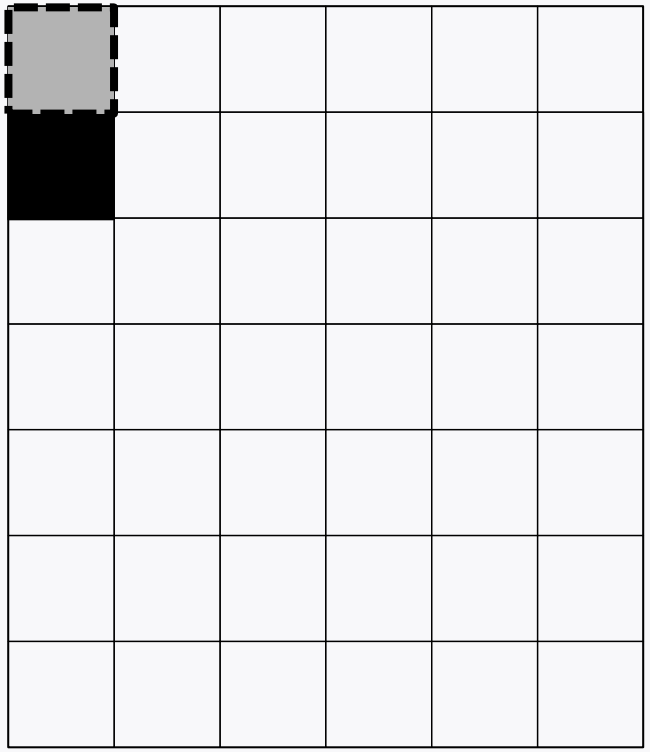
input

conv

What about for next *below* neuron?

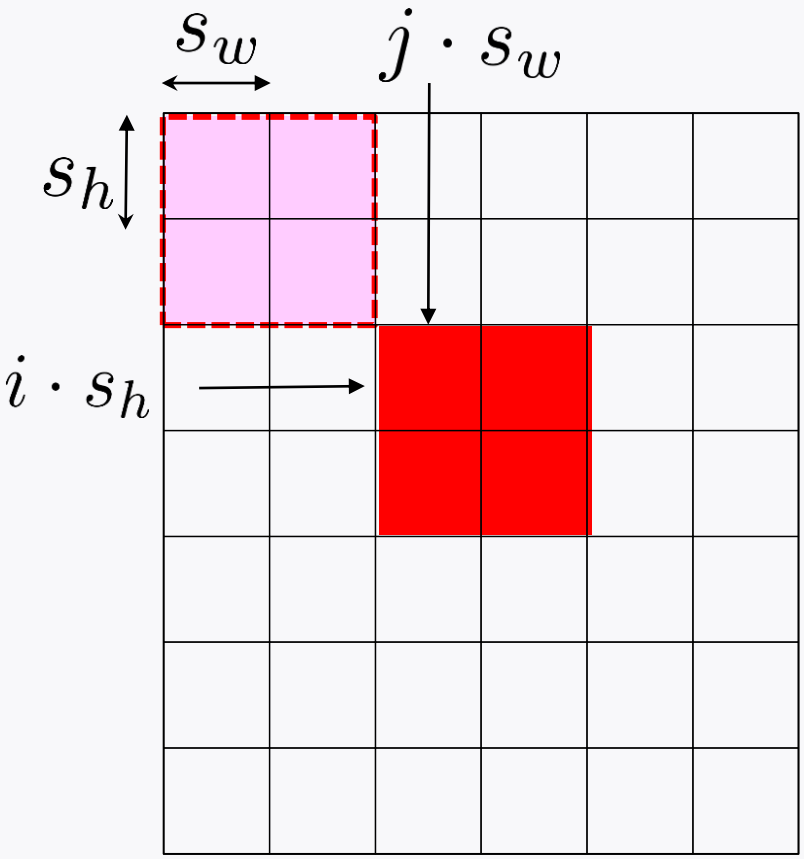


input

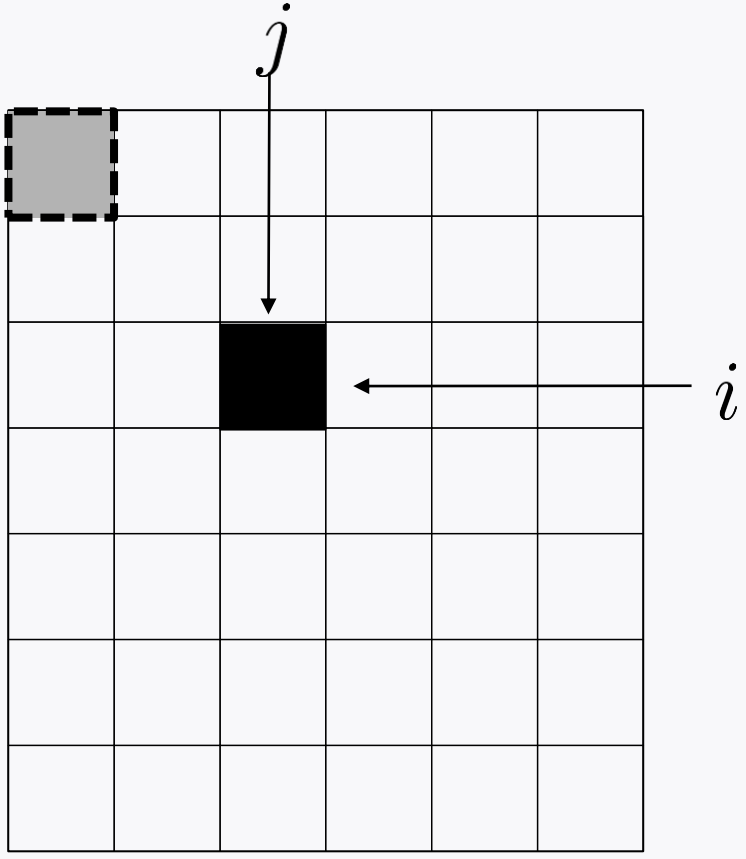


conv

In general …

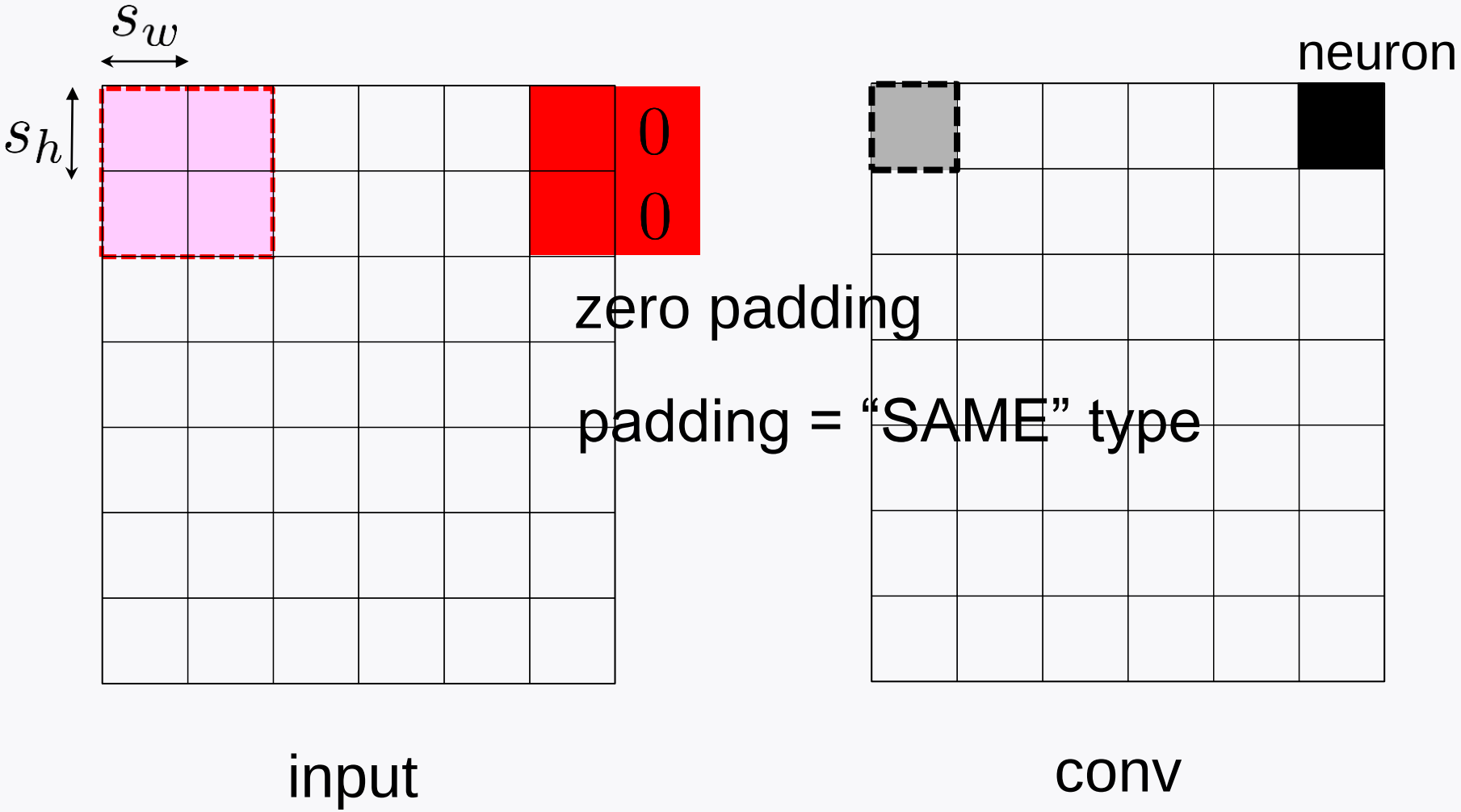


input

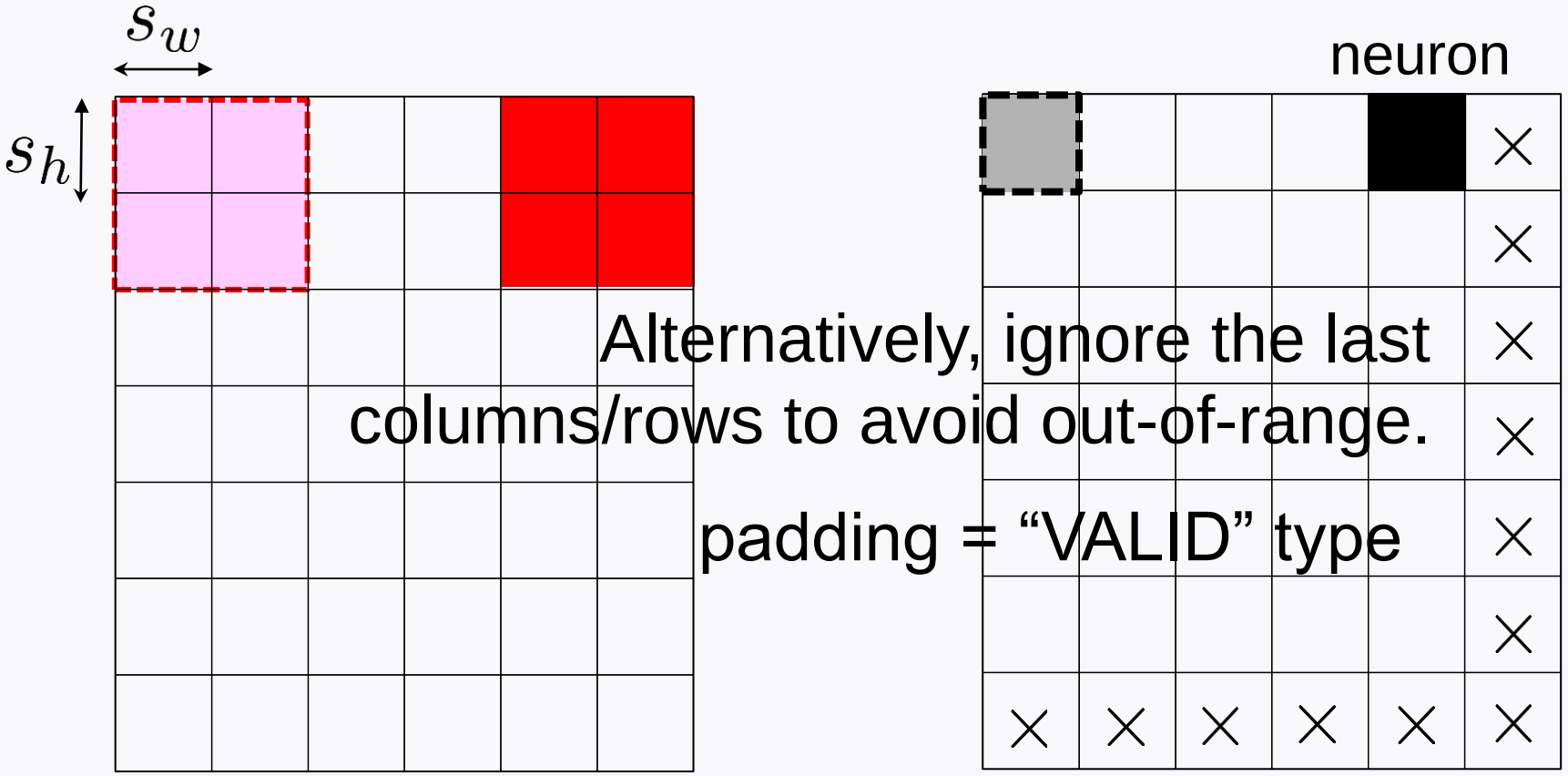


conv

What if a receptive field is out of range?



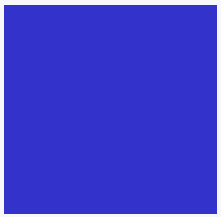
What if a receptive field is out of range?



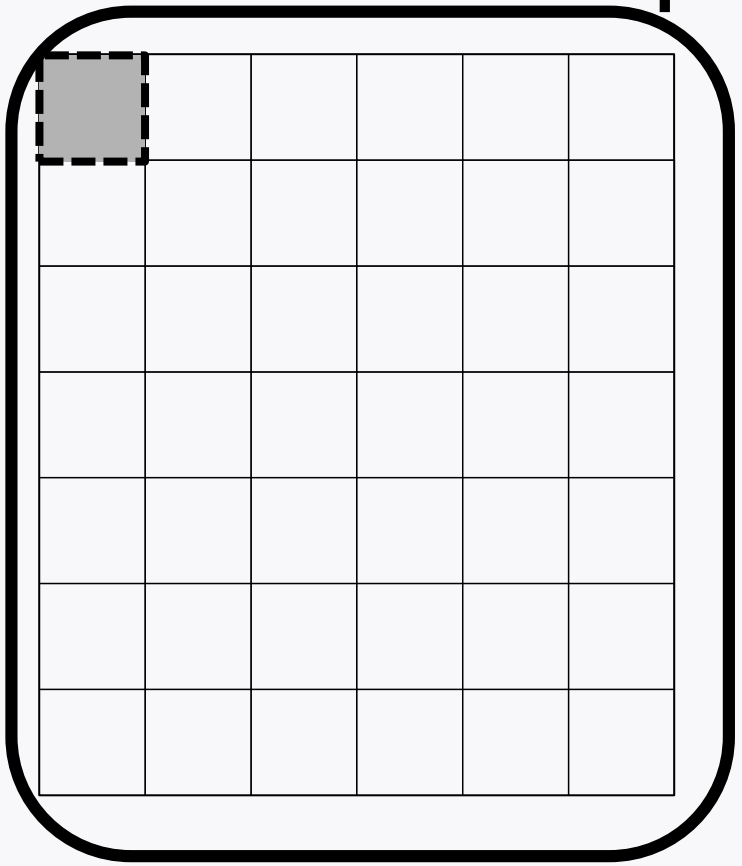
Feature map

Role: Detect a certain pattern (feature).

filter



Called: **feature map**



Note: Filter depends on a target pattern that we wish to detect.

To detect one target pattern, use one filter.

This is why we *use the same filter for a feature map.*

Look ahead

1. Will study further on “feature map”.
2. Will study 2nd building block: **Pooling** layer