Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Convolutional neural networks

Lecture 7

Changho Suh

January 24, 2024

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Convolutional neural networks (CNNs) and history

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

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:

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

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

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

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Recap: DNNs

Model complexity is very high.

This often incurs the overfitting problem.

This motivates the use of generalization techniques:

Regularization; data augmentation;

early stopping; drop out

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Recap: 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
```

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Recap: 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
```

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

What is next?

One important question:

Can DNNs be specialized?

CNNs: Image data

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

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

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.

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Focus of Lecture 7

Focus on CNNs.

Specifically we will:

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

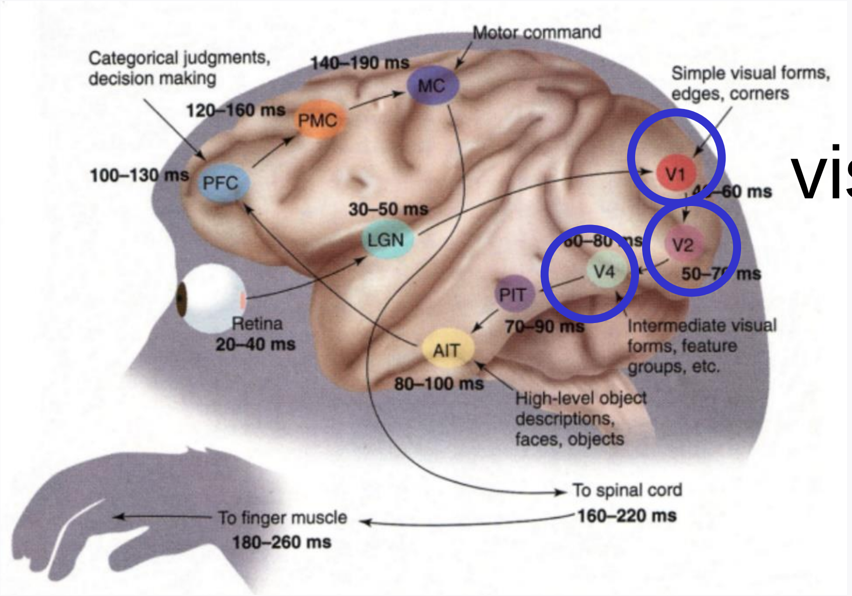
   Conv layer

   Pooling layer

3. Discuss two popular CNN architectures.

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Visual cortex (시각피질)



visual cortex

Hubel & Wiesel ’58/’59 made two observations:

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**.

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

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

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Two building blocks of CNNs

1. Convolutional layer (Conv layer)

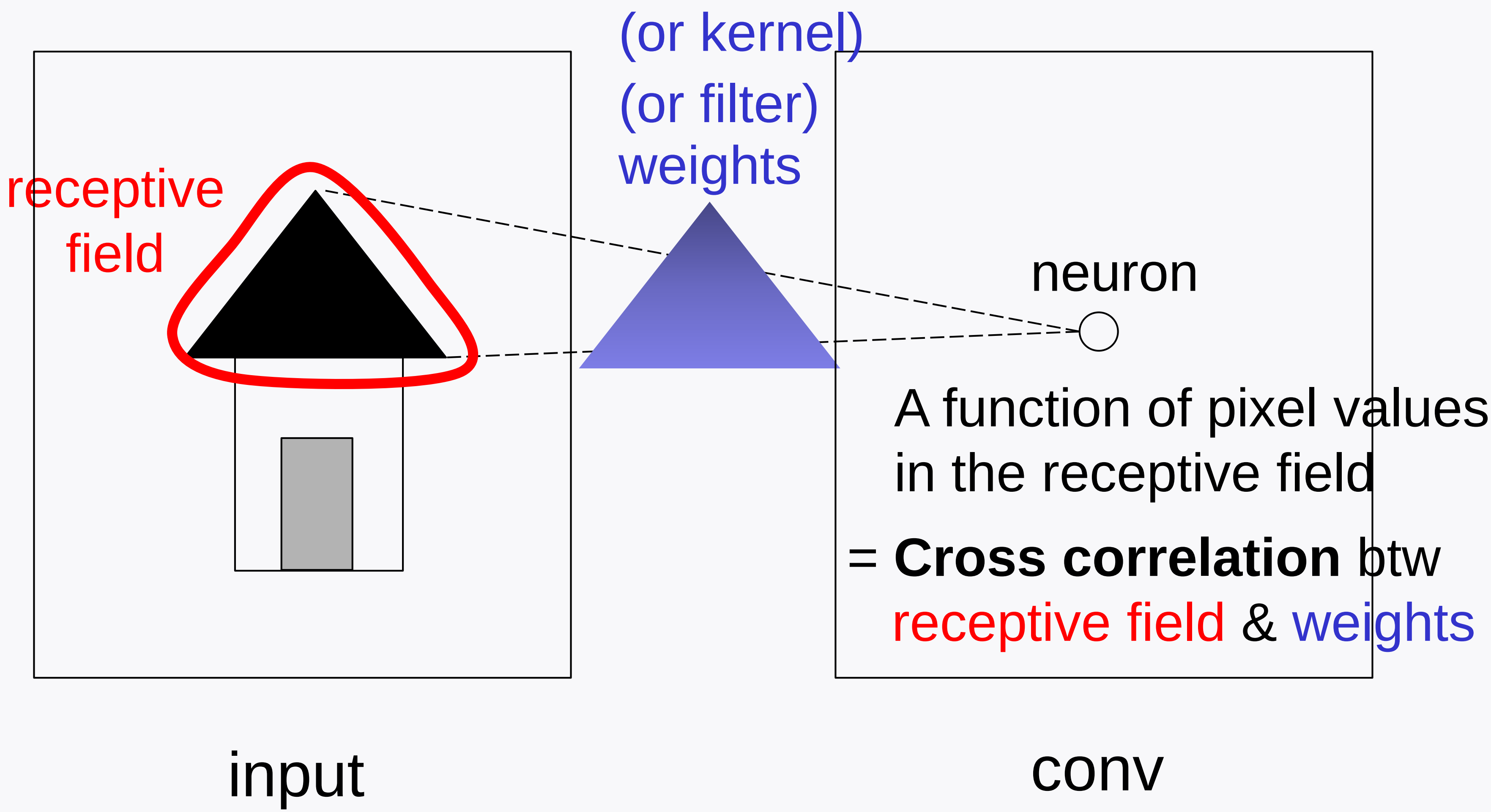
Role: Mimick neurons' behaviors:
Reacting only to receptive fields.

2. Pooling layer

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

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

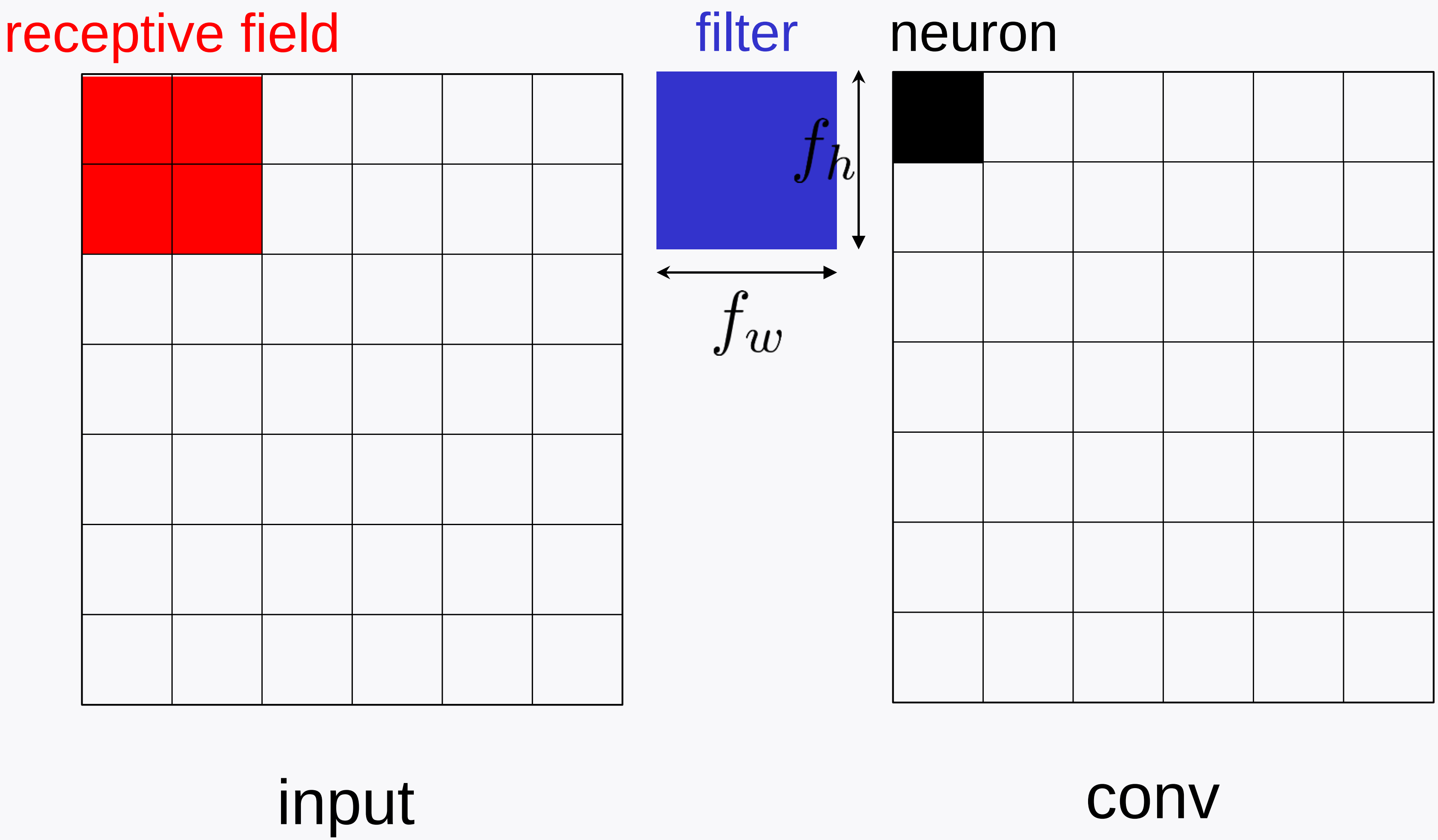
Conv layer



*Convolution operation is very similar to cross correlation.

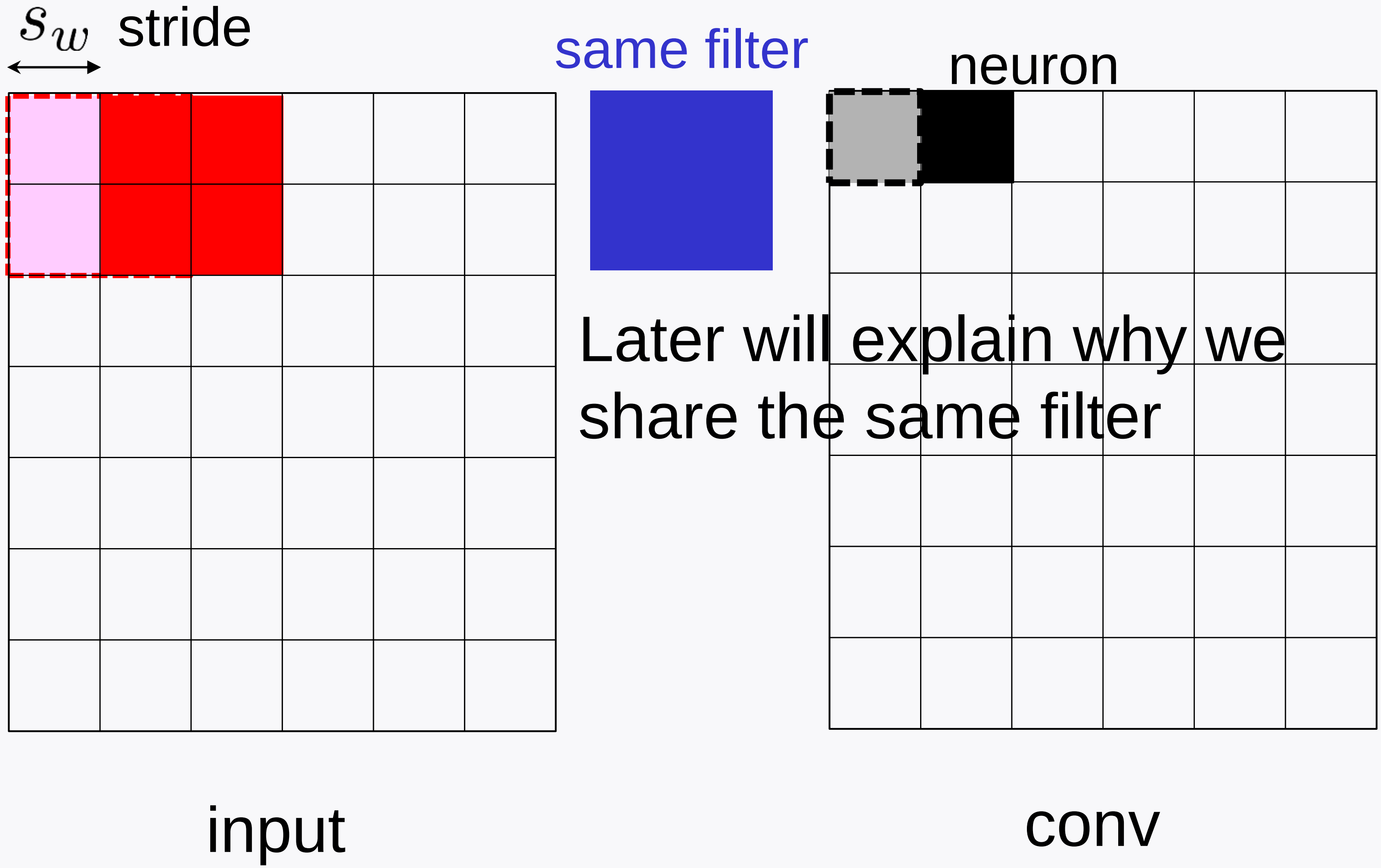
Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Abstraction via grids



Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

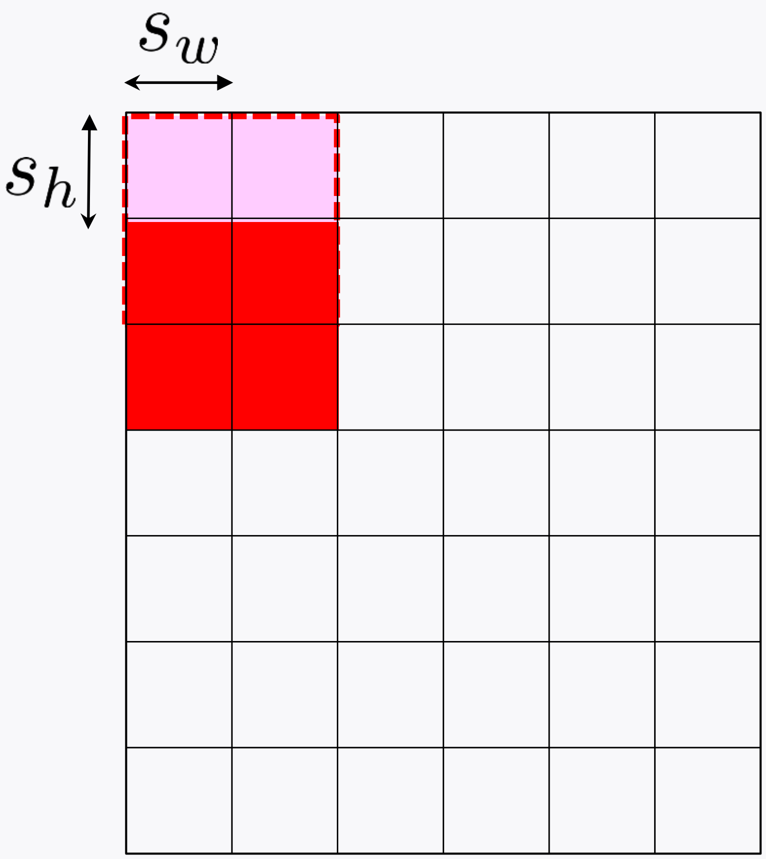
What about for next *right* neuron?



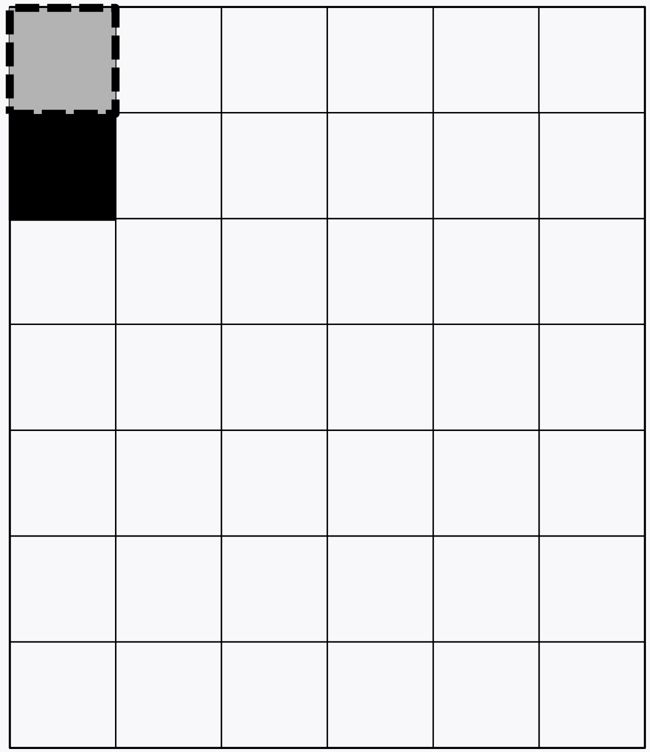
Later will explain why we share the same filter

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

What about for next *below* neuron?



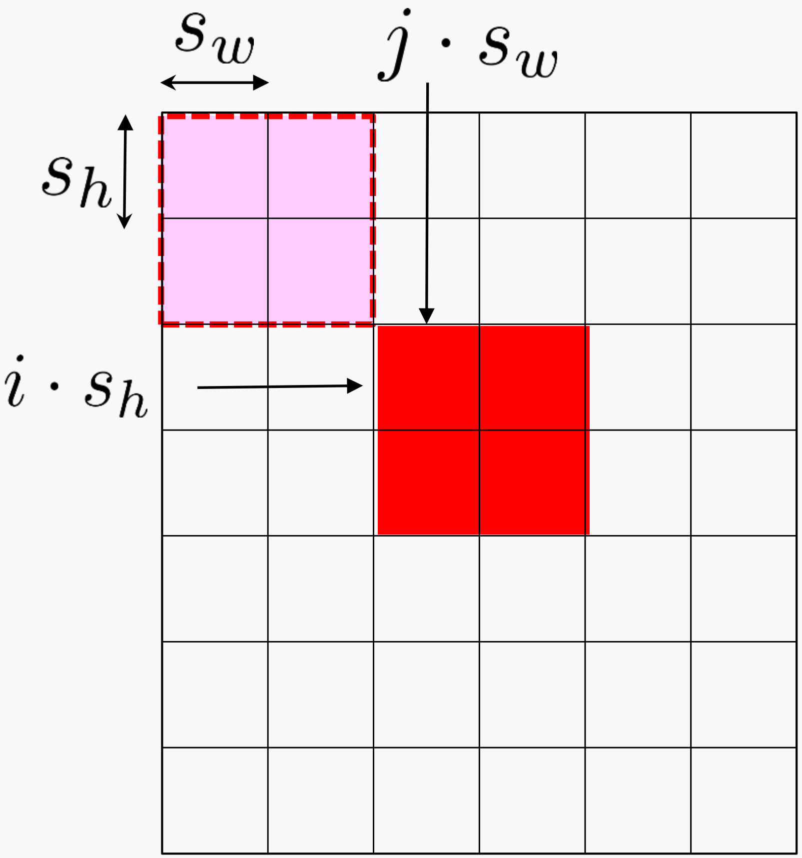
input



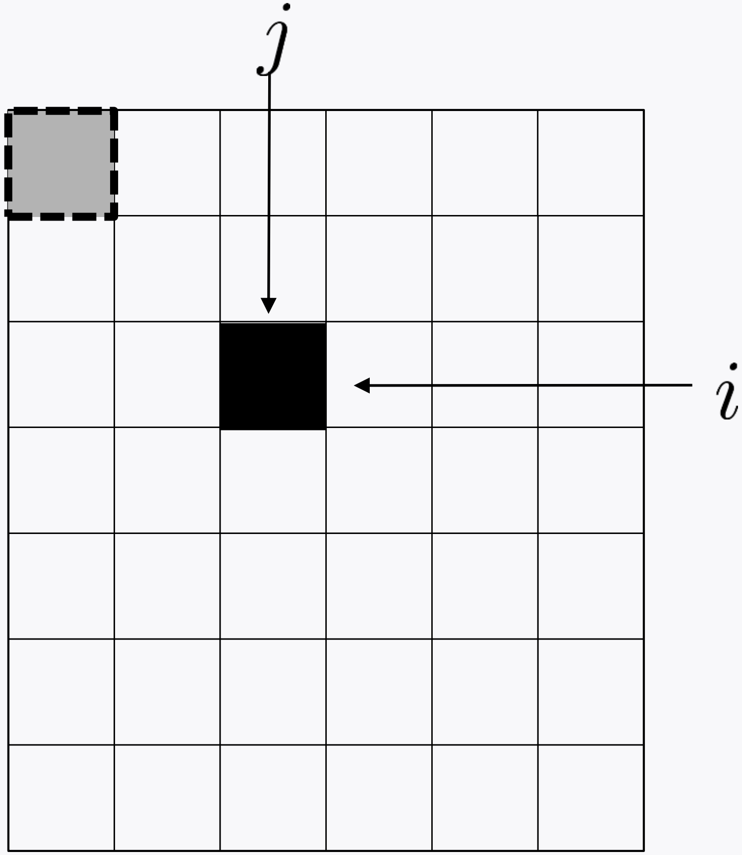
conv

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

In general …



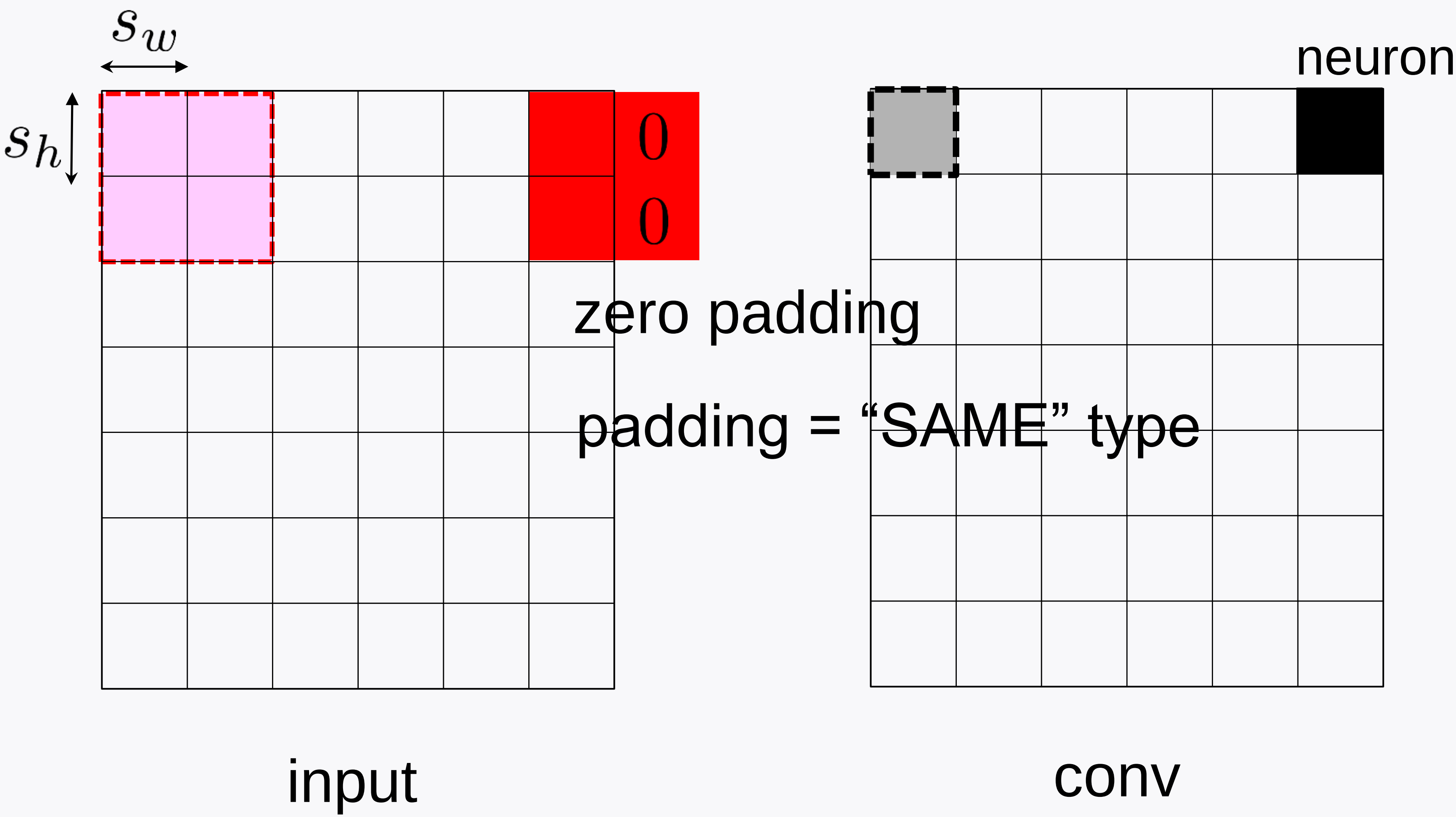
input



conv

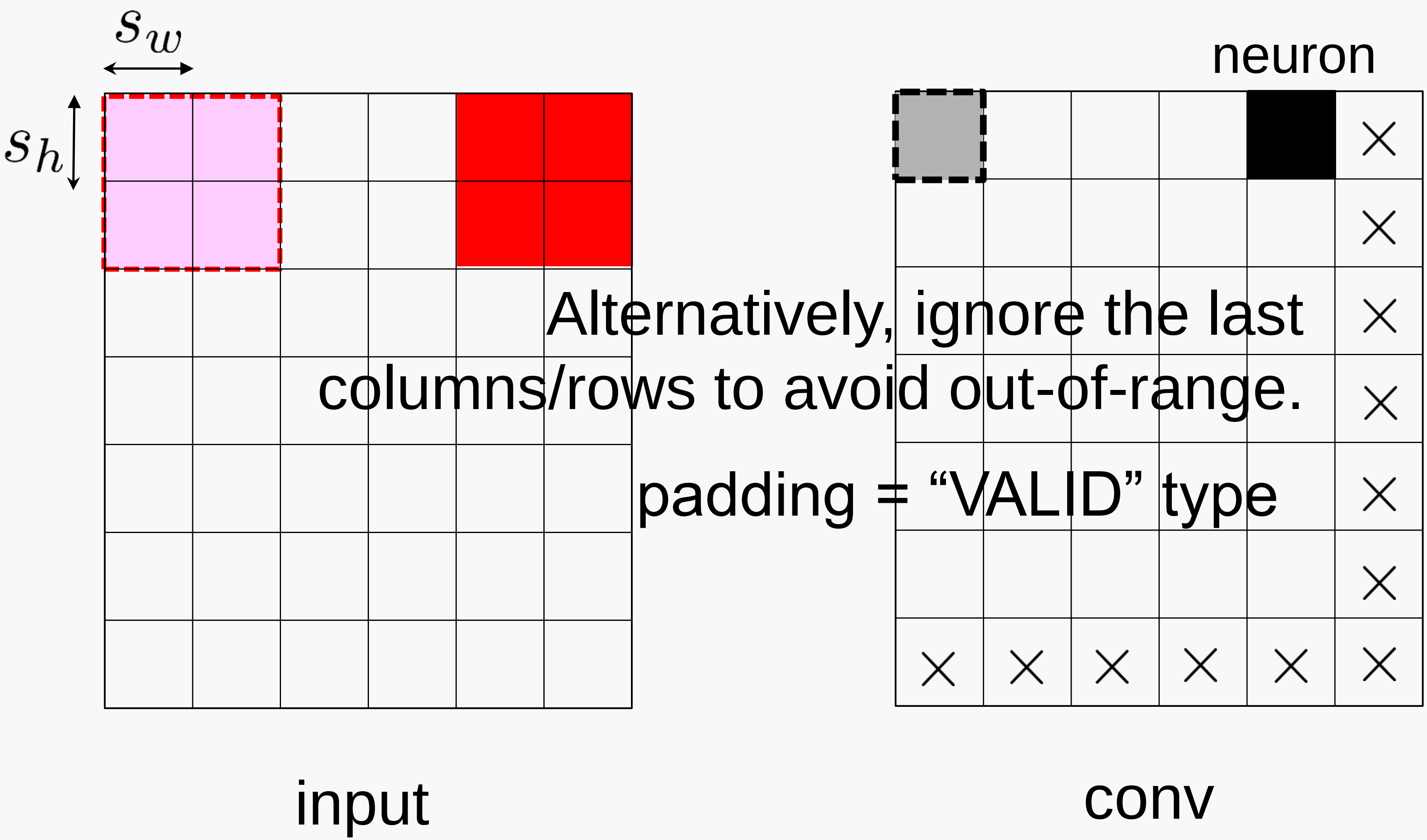
Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

What if a receptive field is out of range?



Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

What if a receptive field is out of range?



Alternatively, ignore the last columns/rows to avoid out-of-range.

padding = "VALID" type

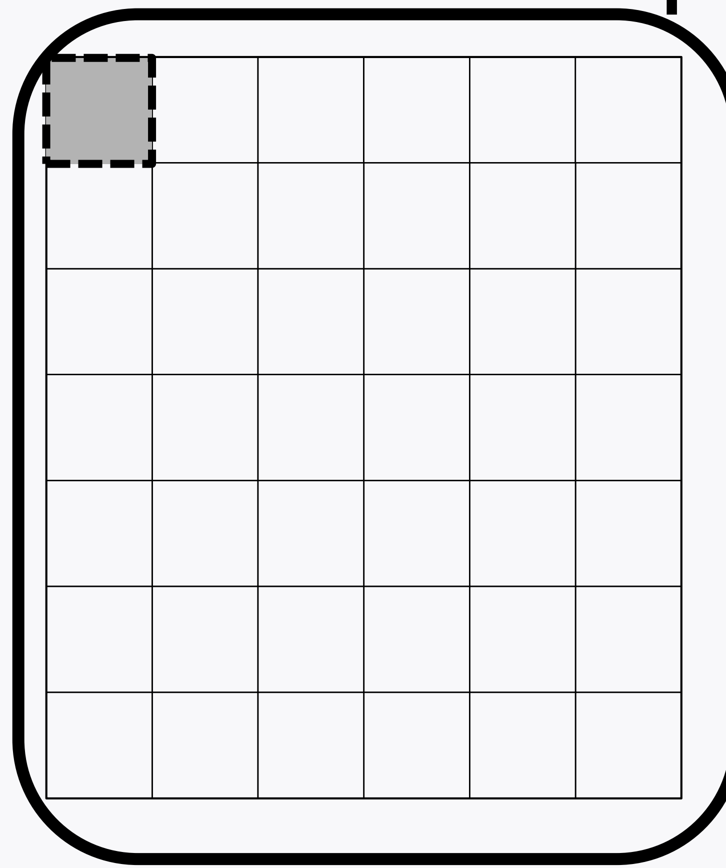
Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

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.*

Copyright © 2024 Changho Suh @ KAIST. All rights reserved.

Look ahead

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