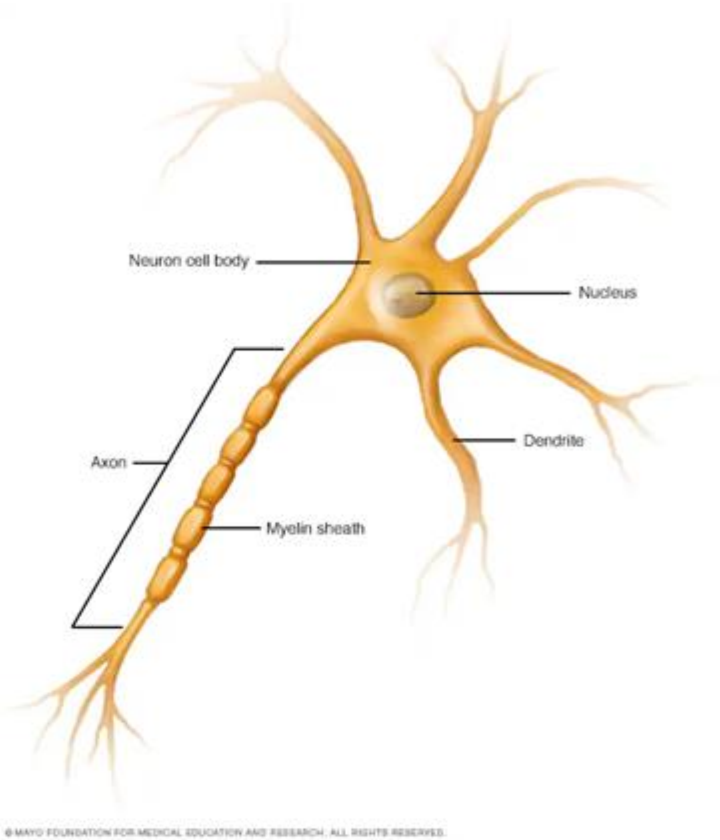
Machine learning concepts: neural networks

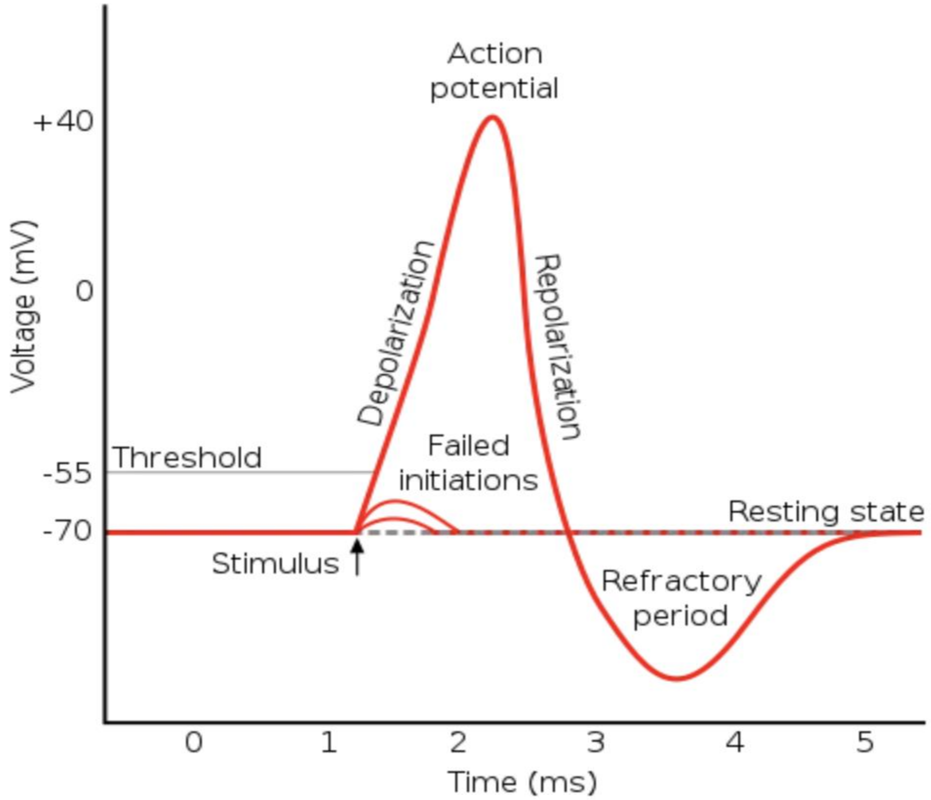
Pranav Rane

Neural Networks

FCNN

- Can learn ANY function regardless number of inputs/outputs
- Modeled after biology…

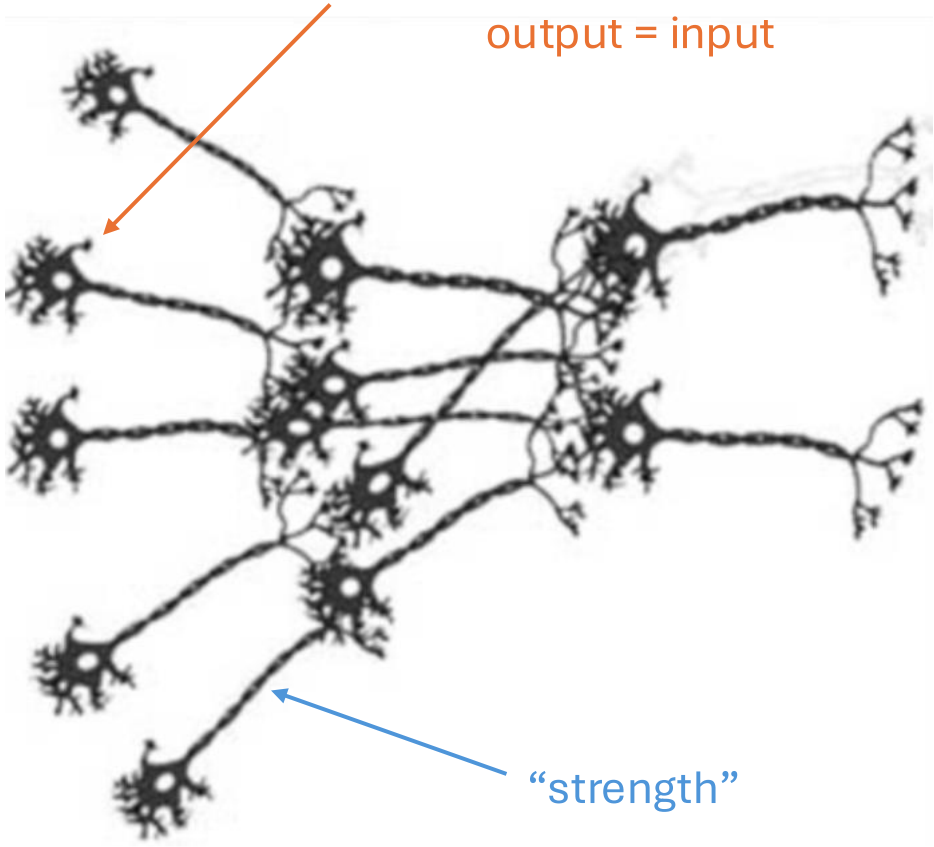


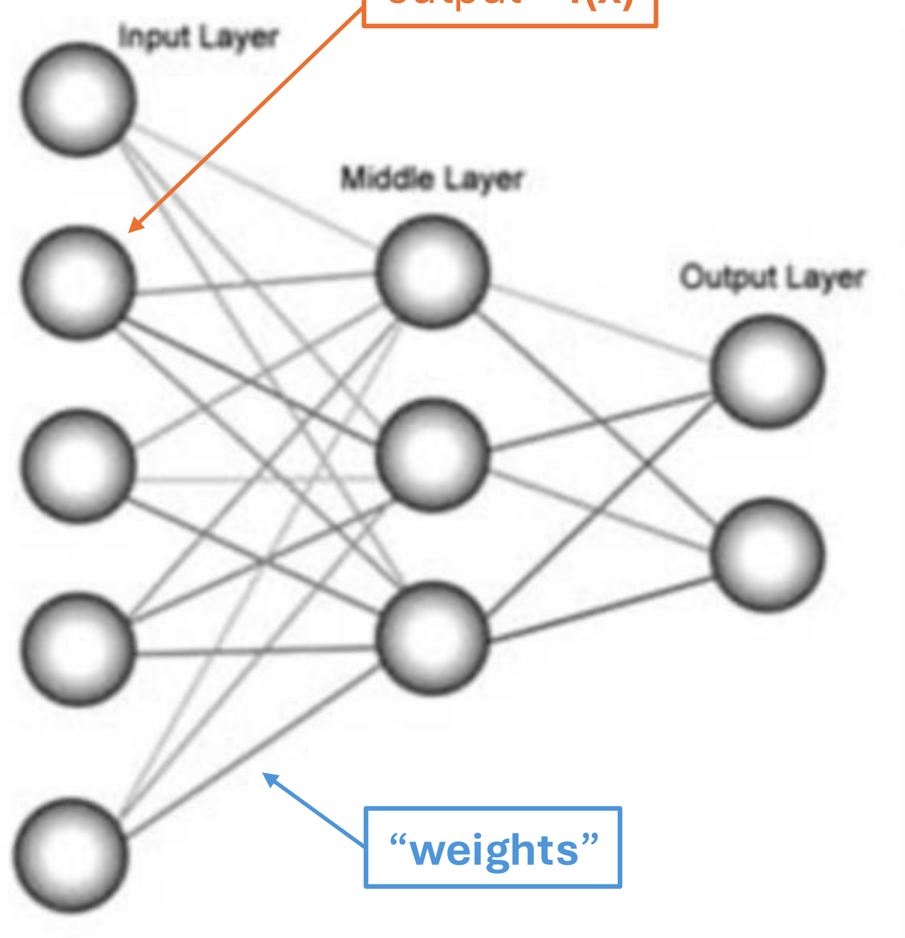


Neural Networks

FCNN

- Can learn <u>ANY</u> function regardless number of inputs/outputs
- Modeled after biology…

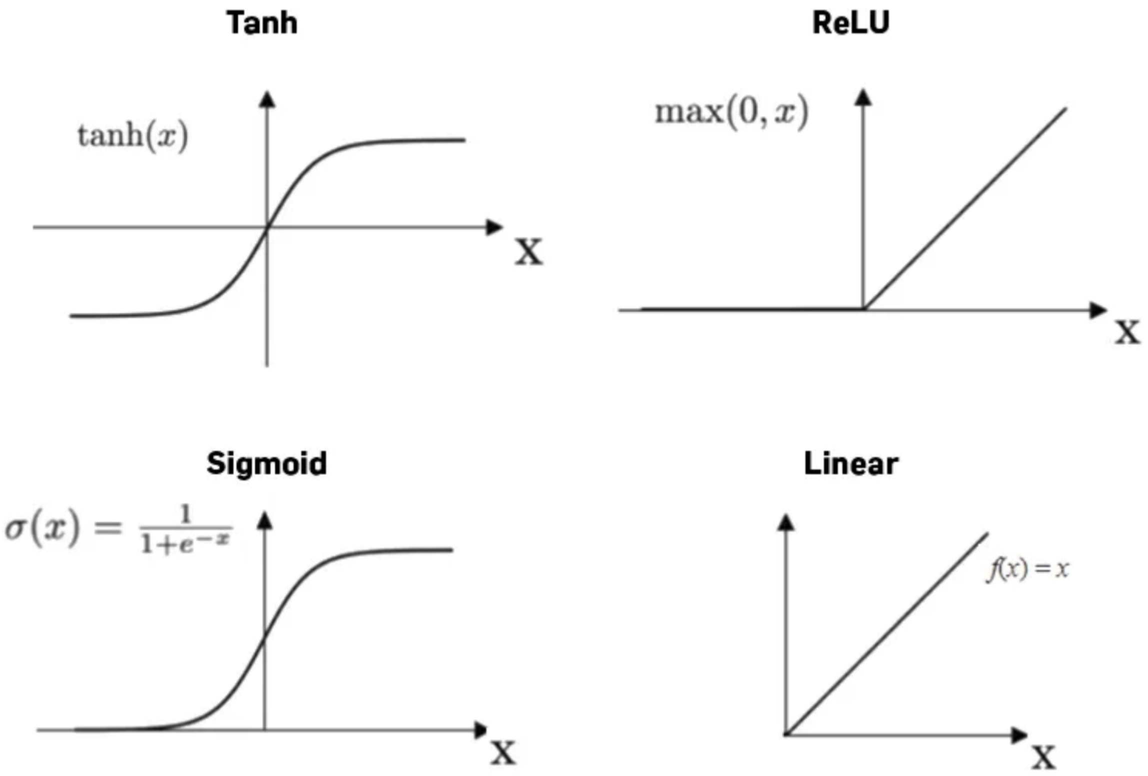




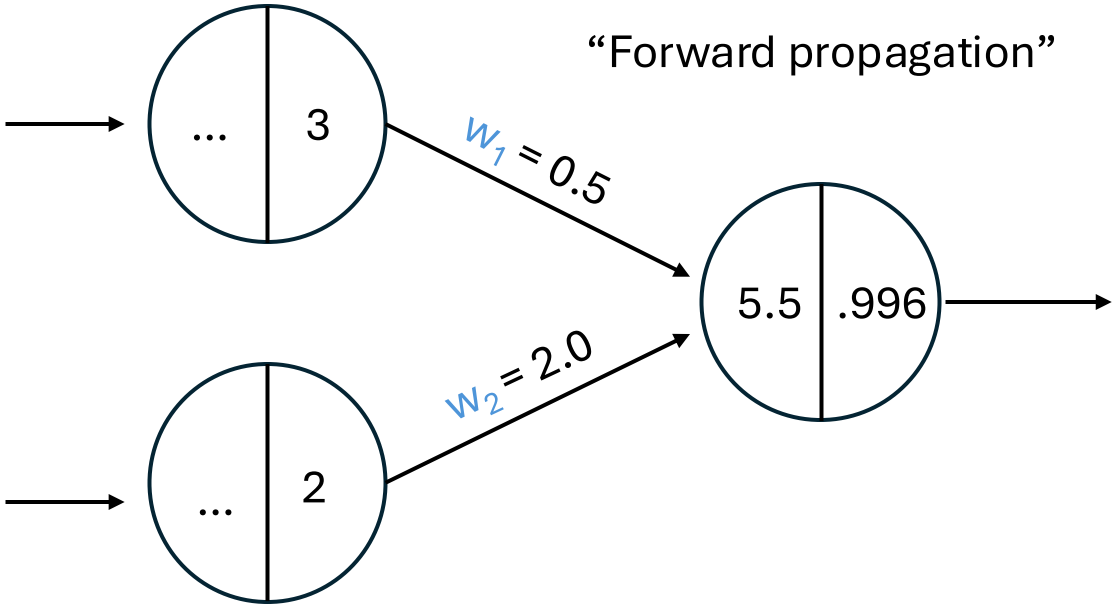
FCNN

f(x) & weights

Activation functions add non-linearity

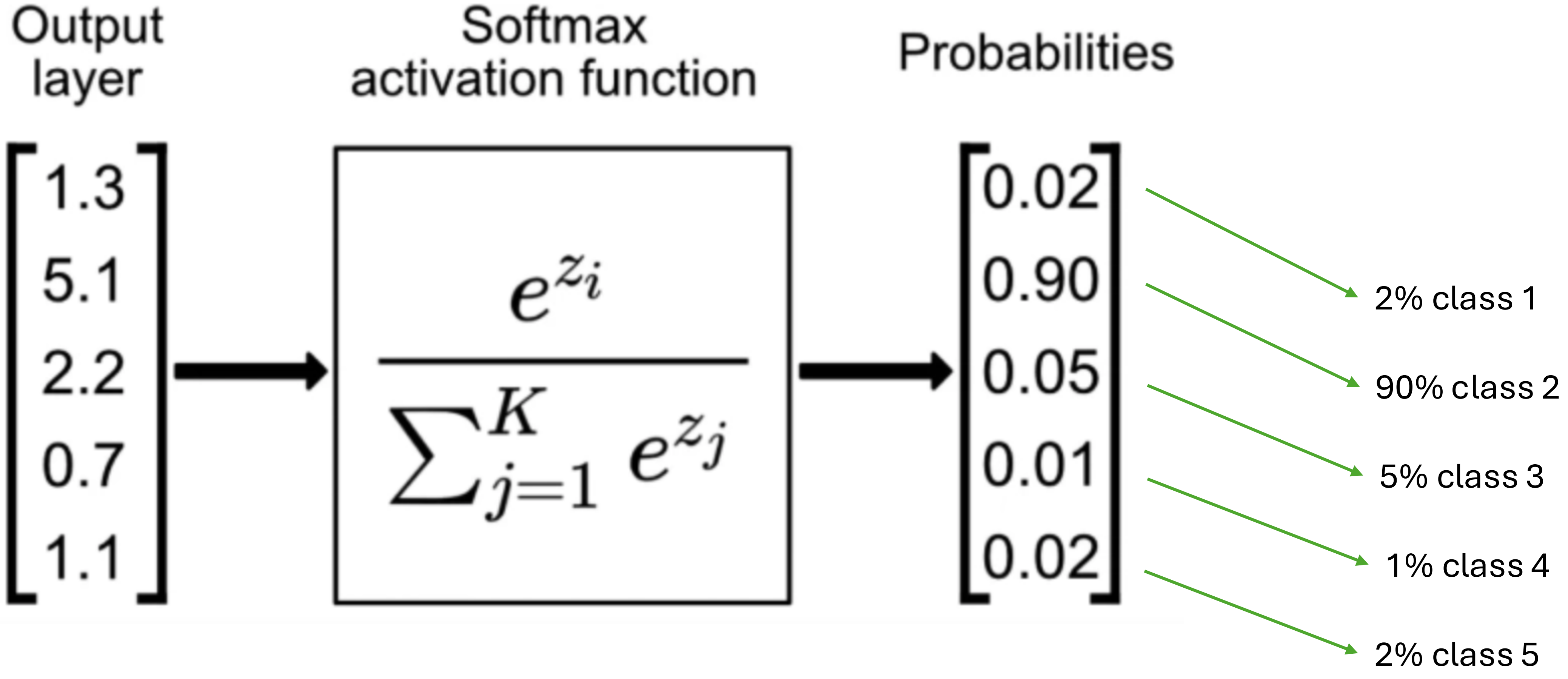


Weights indicate connection strength



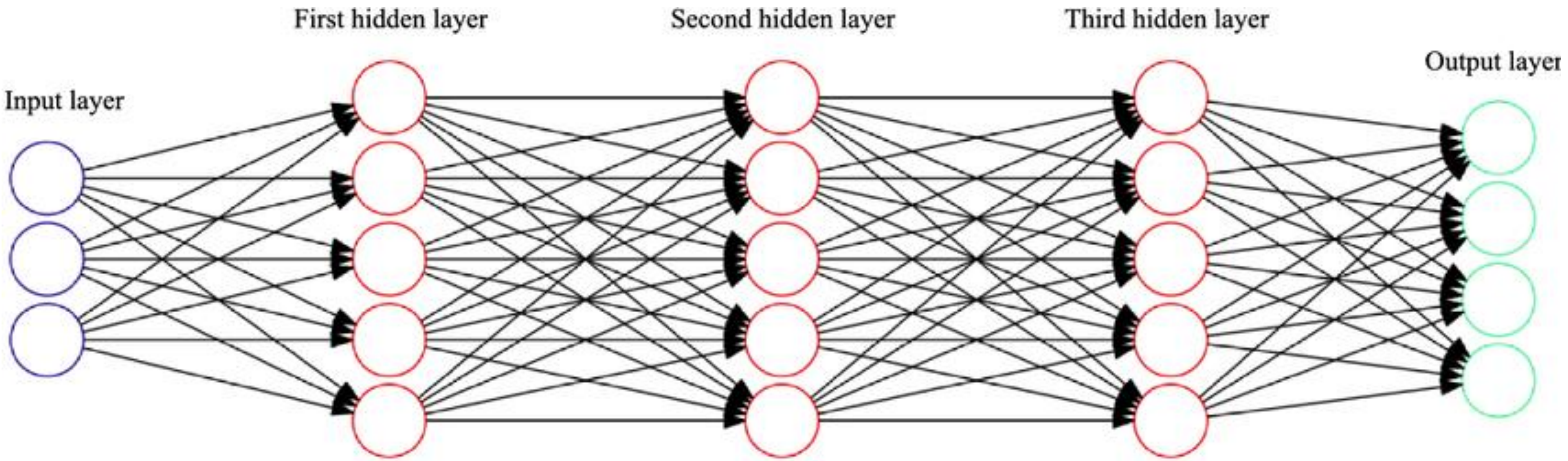
Weights are <u>learned</u> through "Back propagation"

Prediction: softmax activation



FCNN Overview

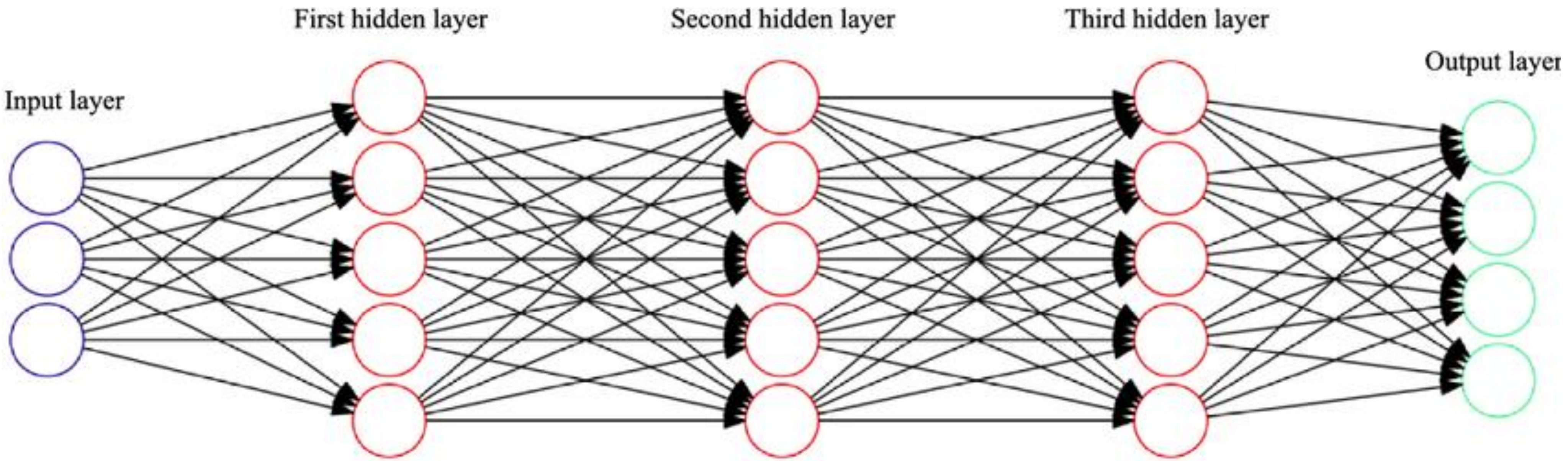
FCNN



1. Choose architecture and activation functions
2. Randomly initialize weights
3. Input training example
4. Forward propagate to reach output layer
5. Compare with training example label, then backpropagate + update weights

Overview

FCNN



1. Choose architecture and activation functions
2. Randomly initialize weights
3. Input training example
4. Forward propagate to reach output layer
5. Compare with training example label, then backpropagate + update weights

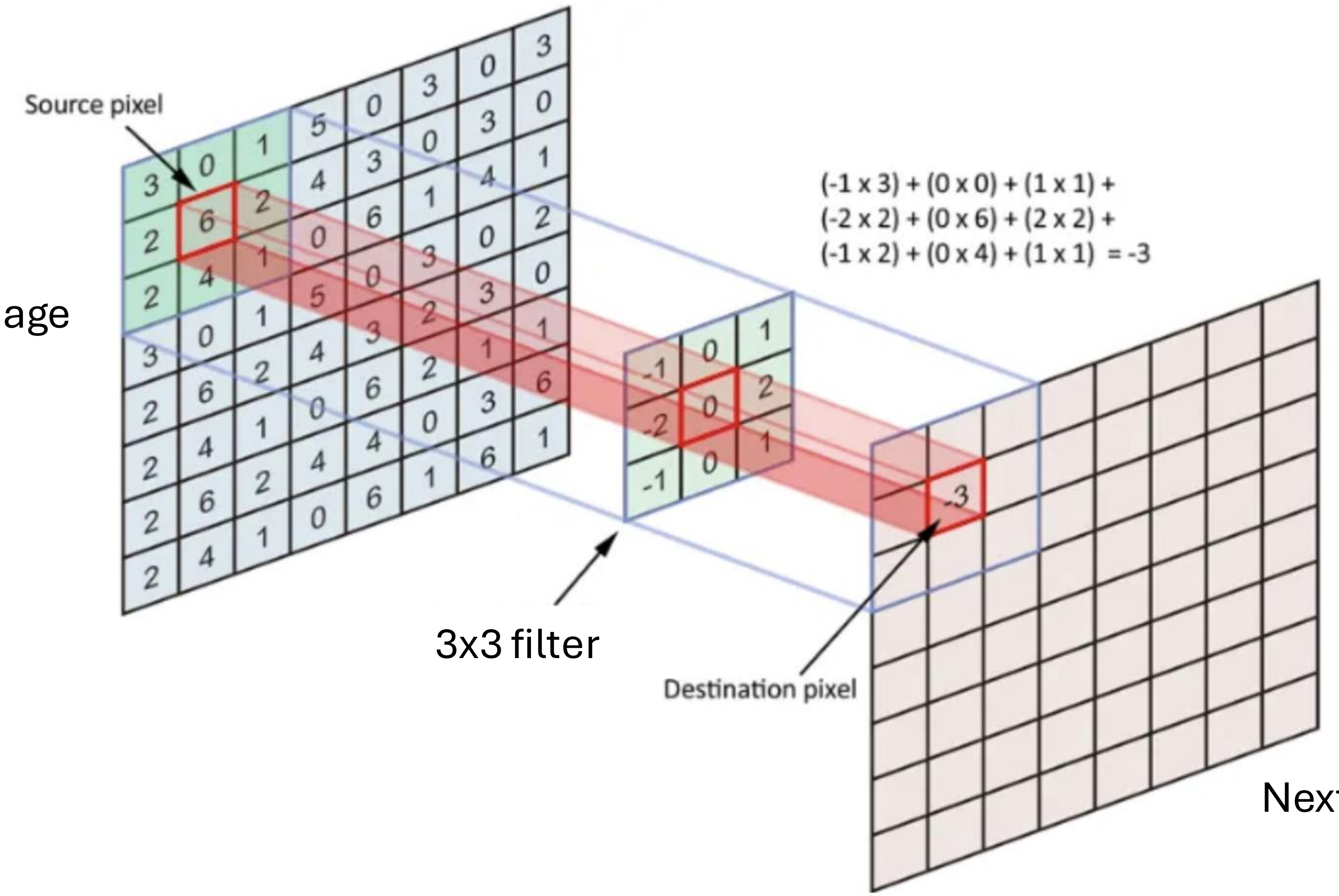
How?

- High model complexity → overfitting
- Trial and error

Convolutional Neural Network

Goal: Understand image data

Architecture: Similar concept, but emphasize spatial position of data

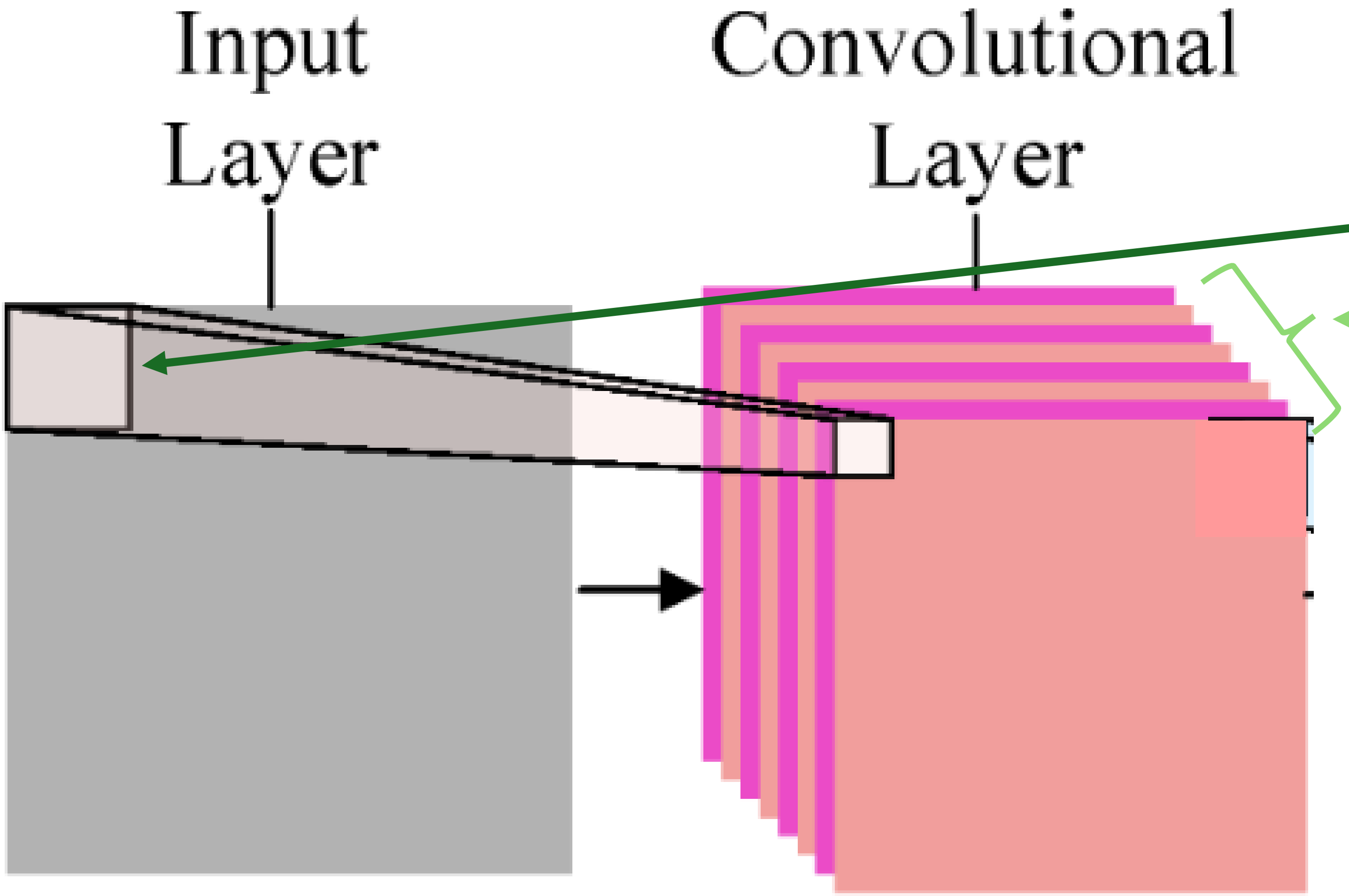


Important concepts:

- Nearby pixels in images are related
- Filters contain weights that are learned during the training process

The meat: convolutional layer

CNN

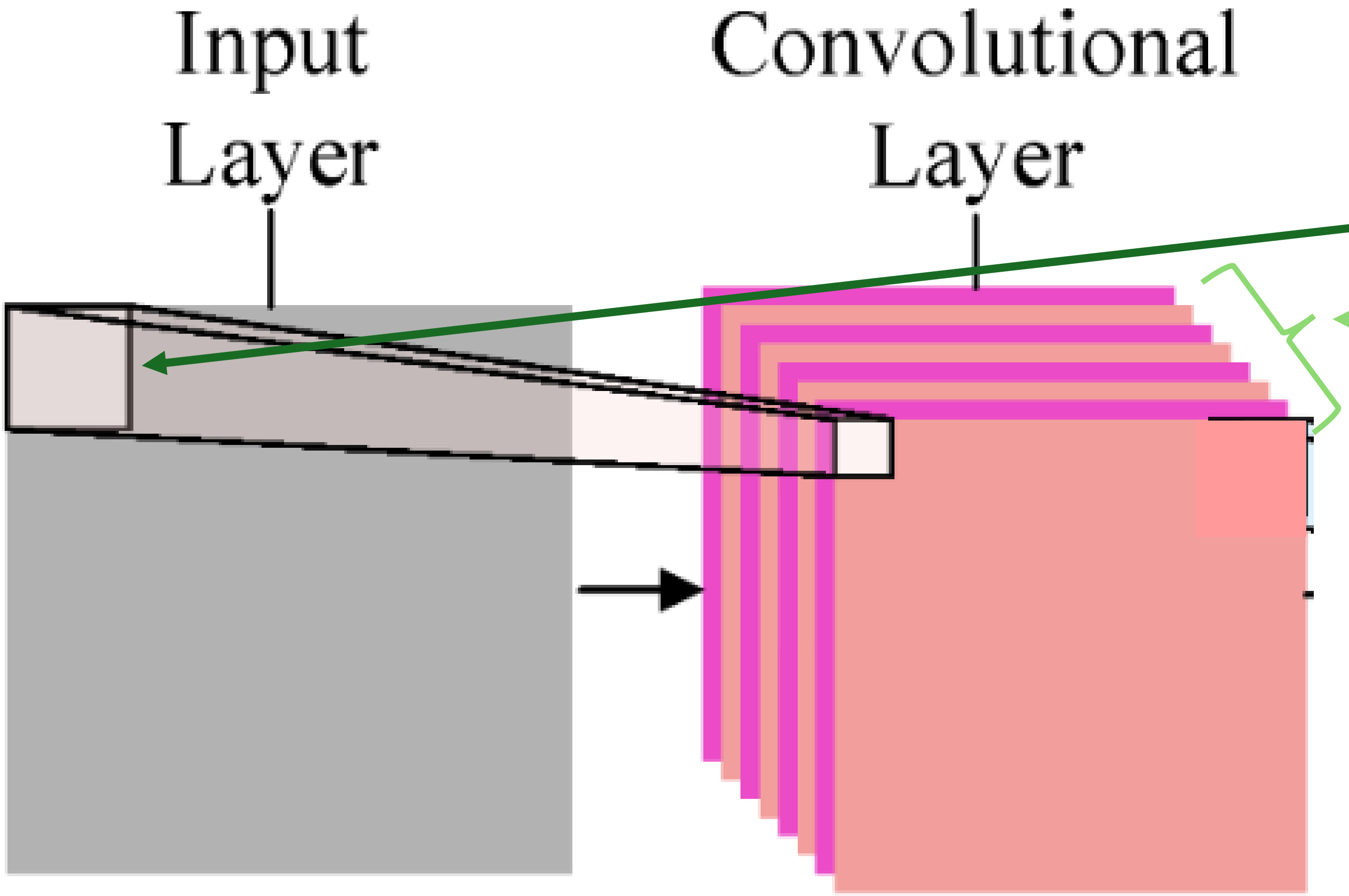


Main convolutional layer parameters:

1. Filter size (3x3,5x5, etc.) adjusts size of receptive field
2. Number of filters (8, 16, 32, etc.) adjusts model complexity
3. Activation function (ReLU) for non-linearity

The meat: convolutional layer

CNN



Main convolutional layer parameters:

1. Filter size (3x3,5x5, etc.) adjusts size of receptive field
2. Number of filters (8, 16, 32, etc.) adjusts model complexity
3. Activation function (ReLU) for non-linearity

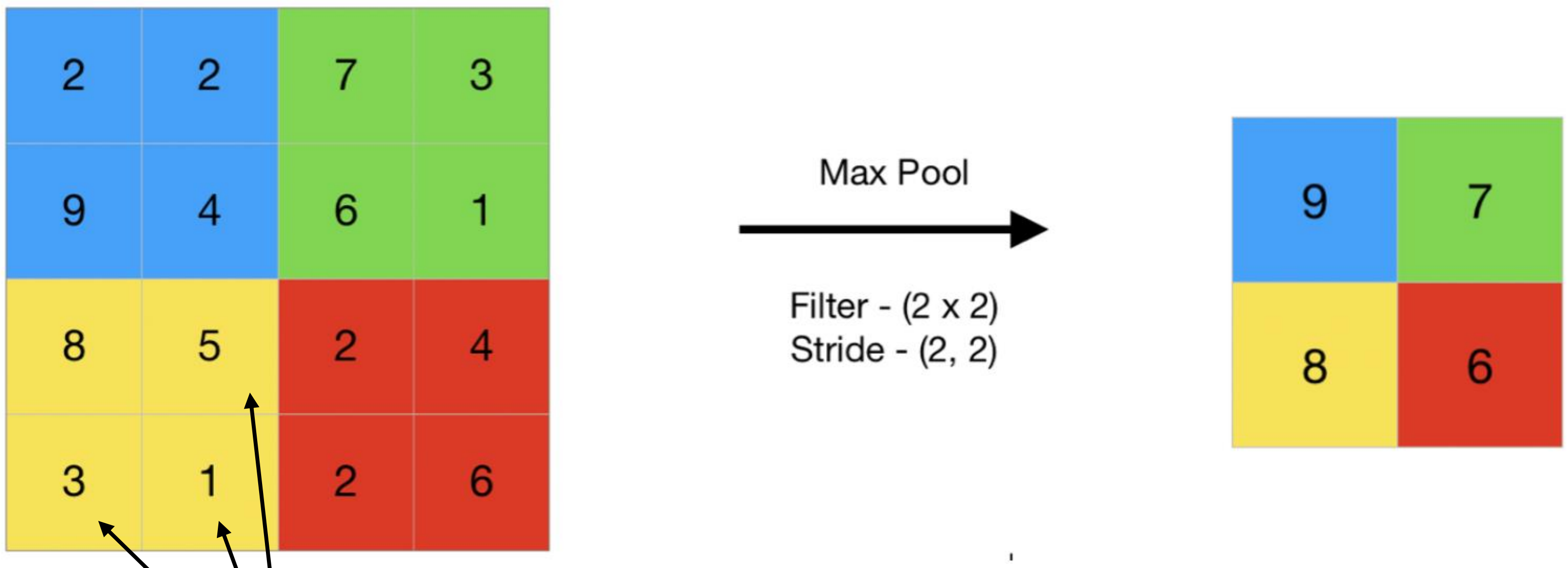
Important concept

Applying multiple convolution layers one after another allows hierarchical learning of more complex patterns/features in the image (ex. edges→ triangles→bird beak)

Max Pooling Layer

- Not super important but it appears in almost every diagram of a CNN
- Reduce dimensionality + keep important features after convolution

Output from convolution layer

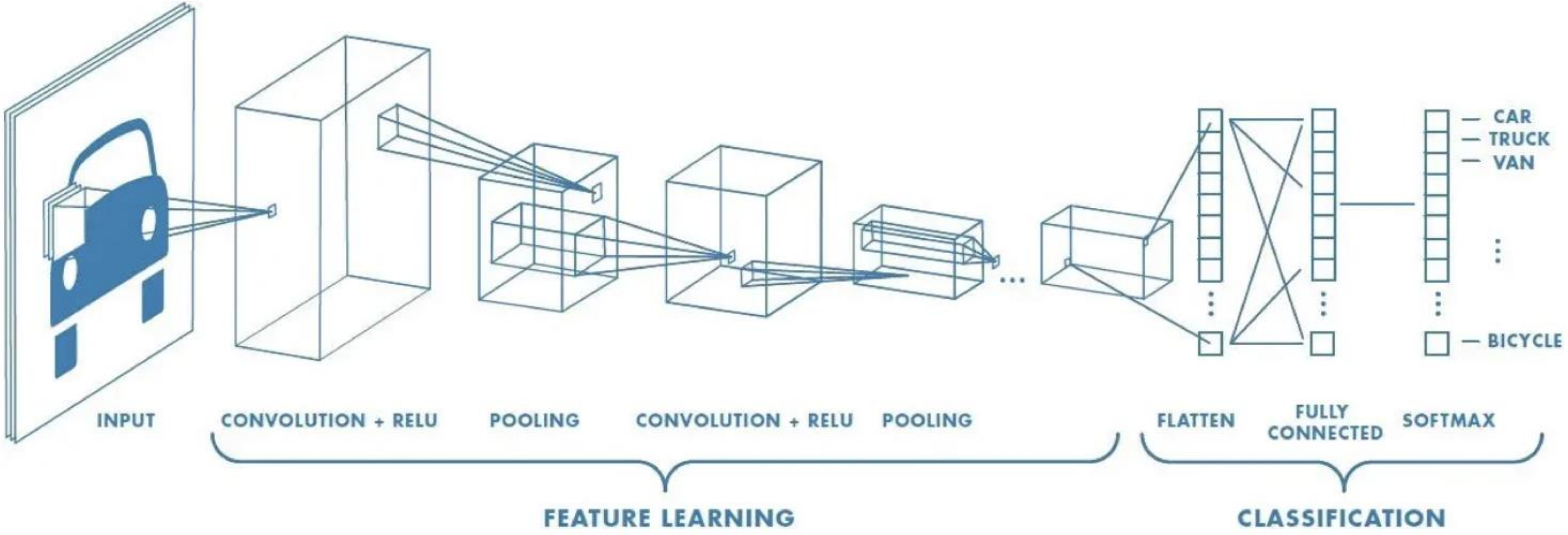


Throws away low values
to keep highest feature

<u>Main pooling layer parameters:</u>

1. Filter size (2x2, 4x4, etc)
2. Stride (1,2,etc.)

CNN Overview

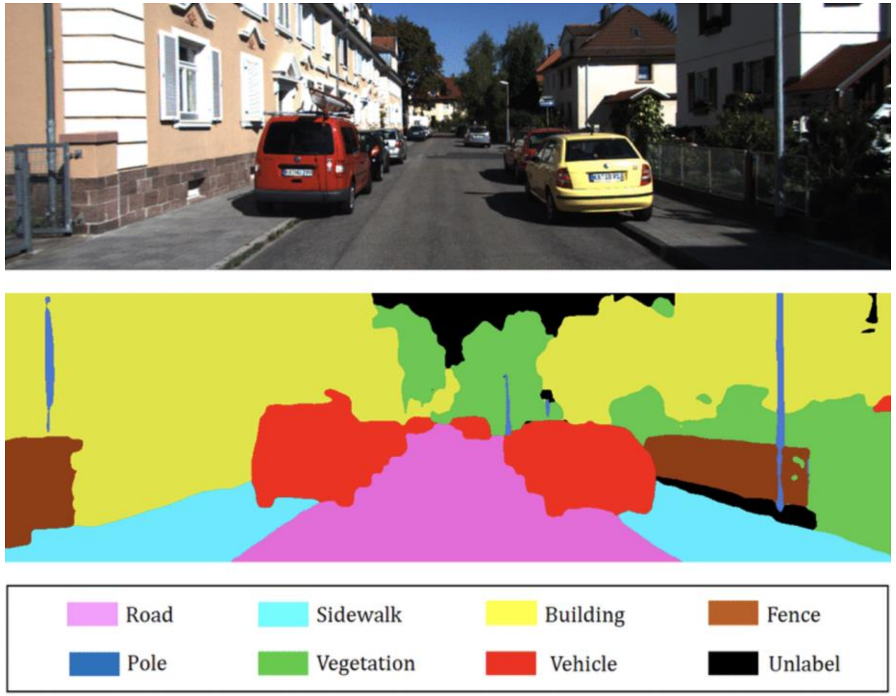


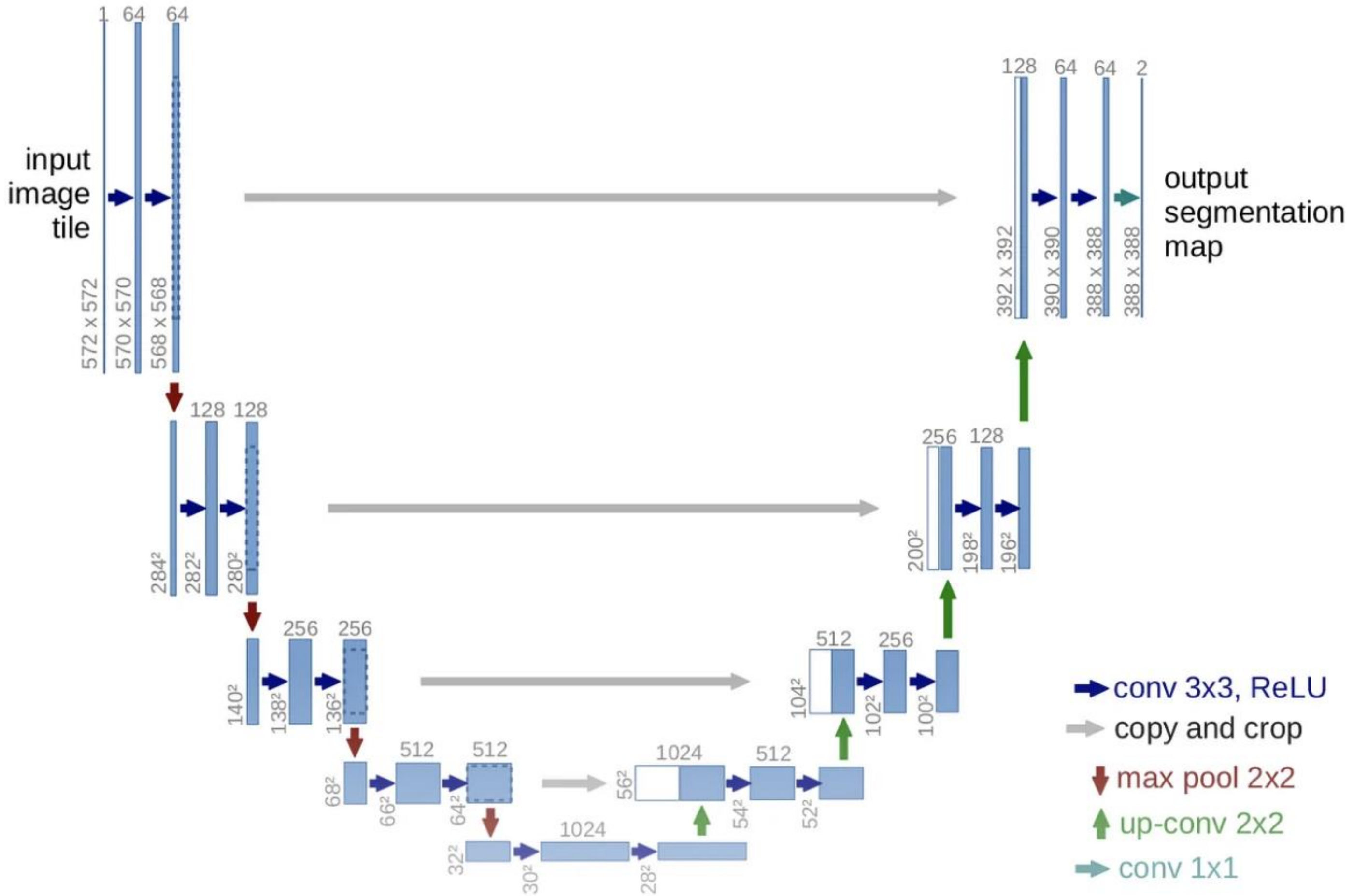
Semantic vs. Instance segmentation

CNN

Semantic segmentation (SS) is pixel classification

SS state of the art uses U-net architecture (2015) that features an encoder/decoder





UNet architecture

Semantic vs. Instance segmentation

CNN

Semantic segmentation (SS) is pixel classification
SS state of the art uses U-net architecture (2015) that features an encoder/decoder
Instance segmentation is pixel classification + distinguishing different instances

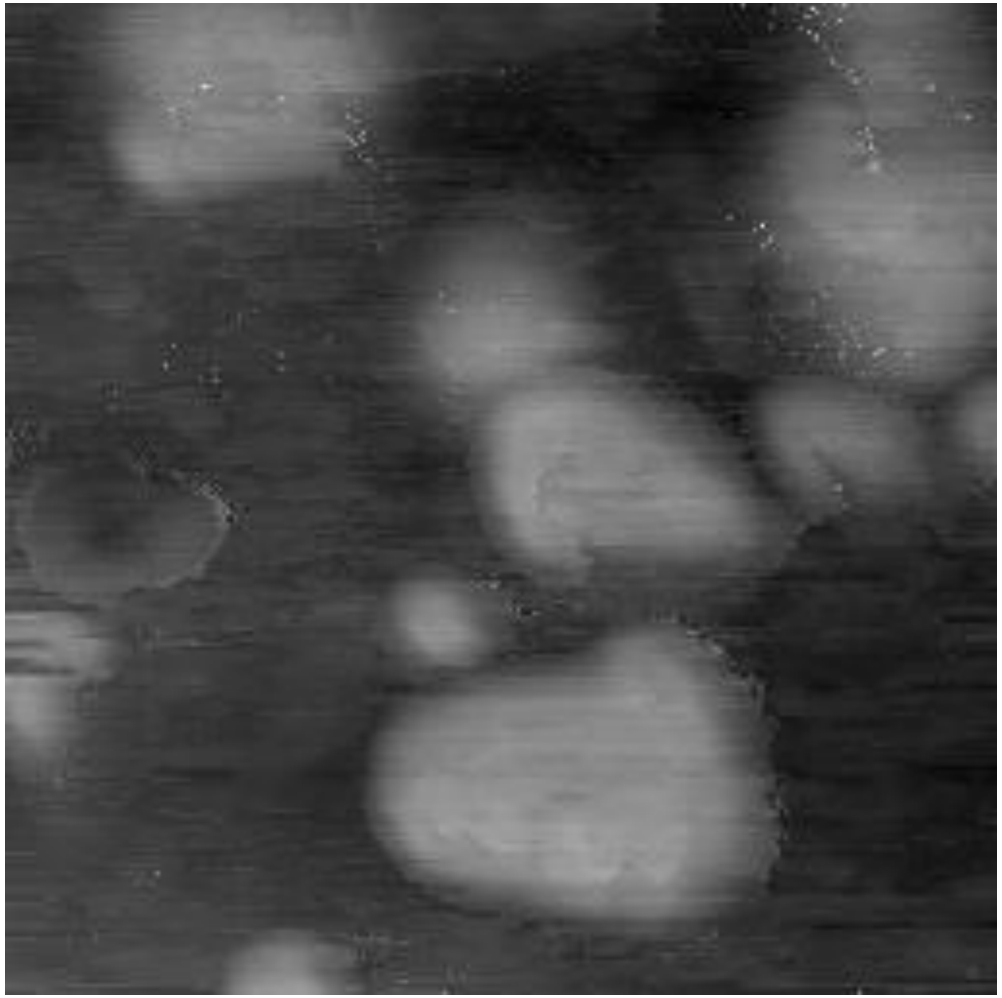
semantic

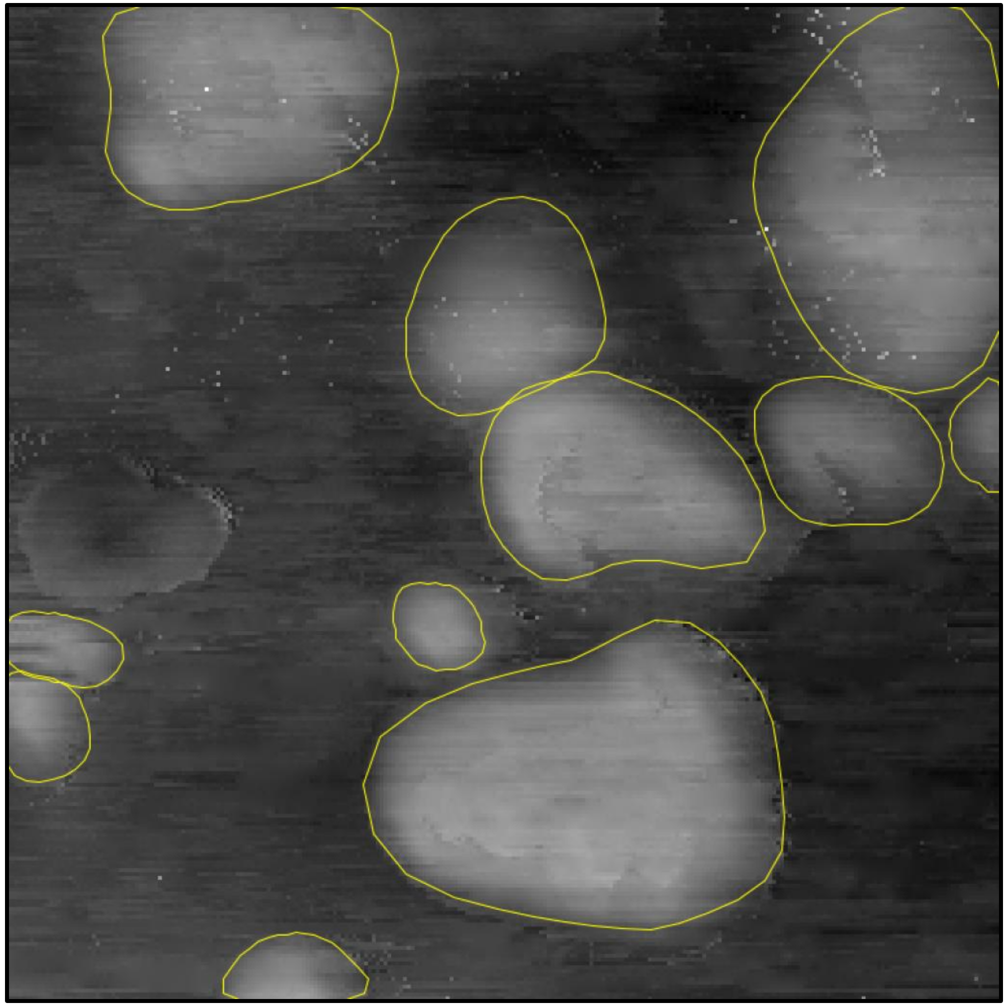
instance

Stardist

StarDist

- Want to identify *individual* blisters accurately (instance segmentation)

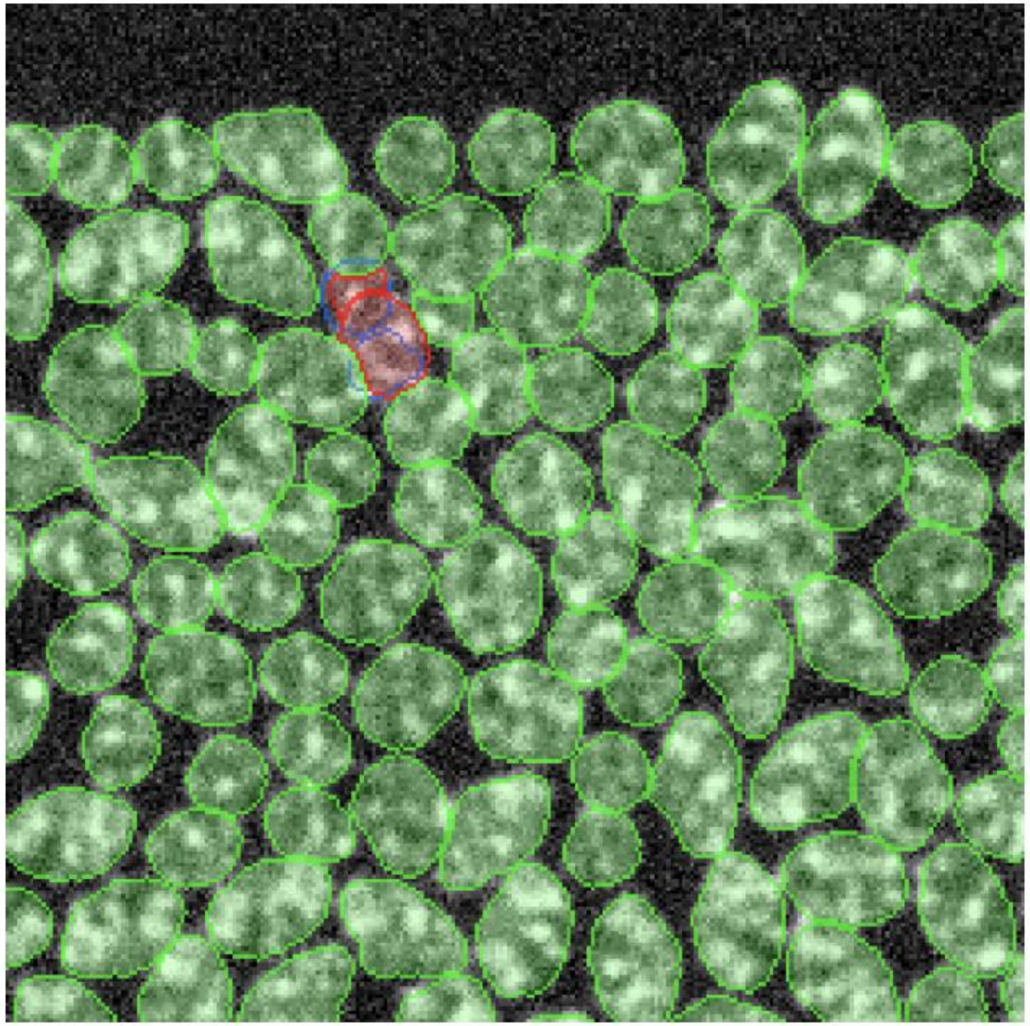




Stardist

StarDist

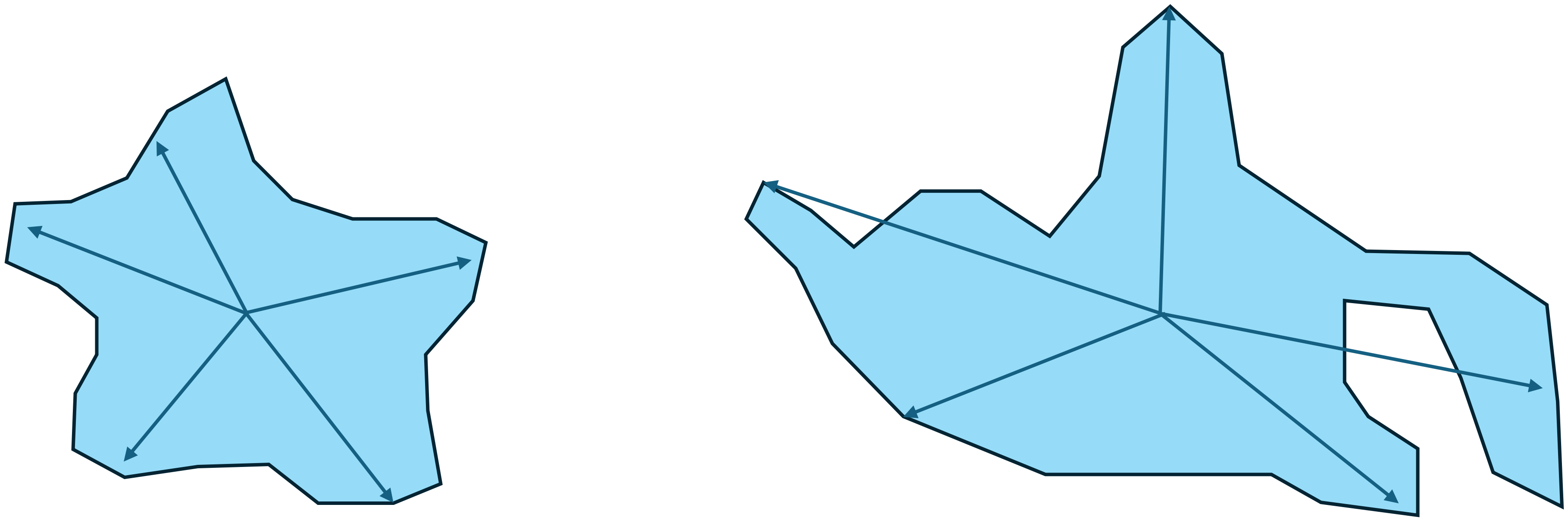
- Want to identify *individual* blisters accurately (instance segmentation)
- Developed to provide better instance segmentation in biological images



Stardist

StarDist

- Want to identify *individual* blisters accurately (instance segmentation)
- Developed to provide better instance segmentation in biological images
- Great for segmenting star-convex polygons



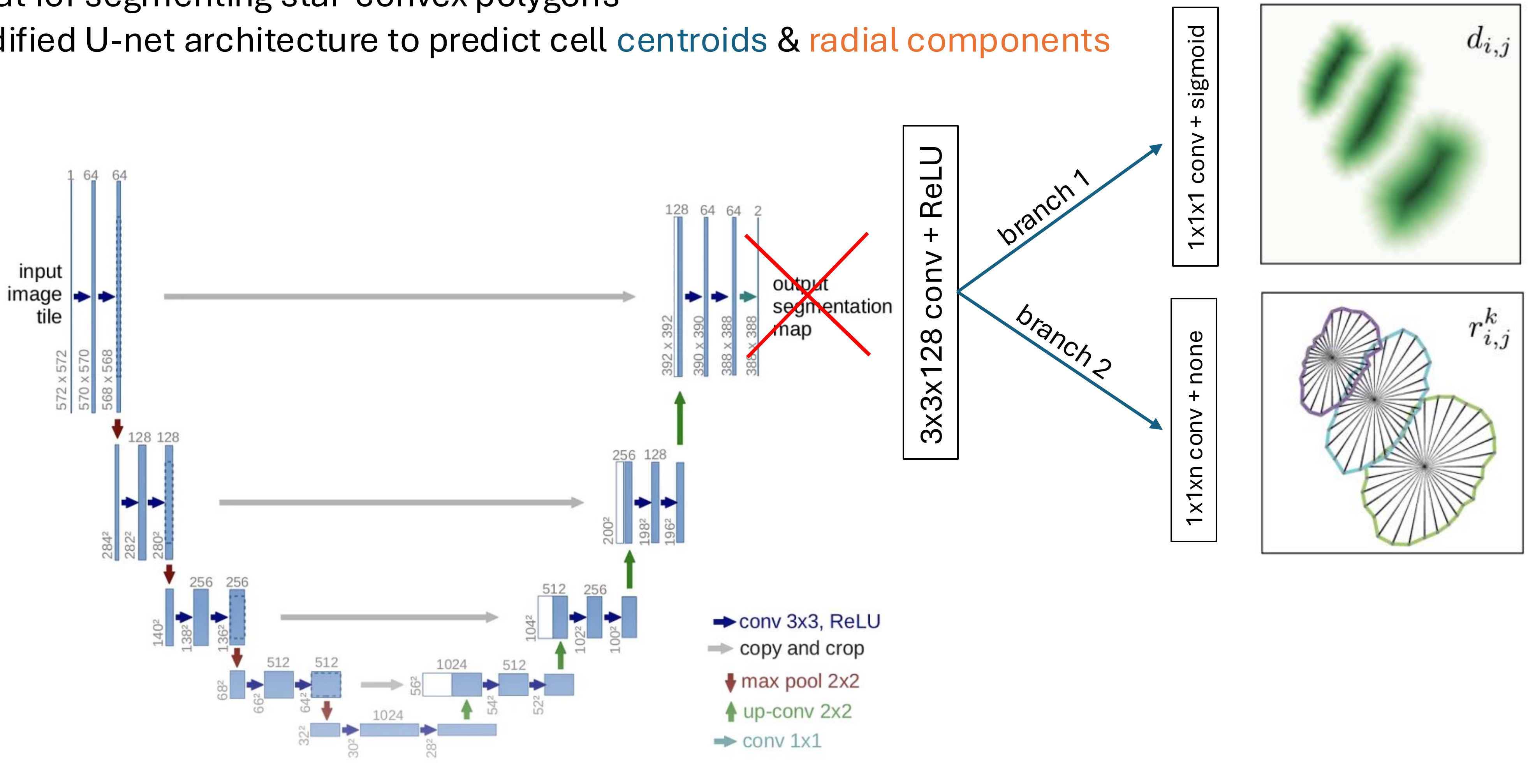
Star convex

Not star convex

StarDist

Stardist

- Want to identify *individual* blisters accurately (instance segmentation)
- Developed to provide better instance segmentation in biological images
- Great for segmenting star-convex polygons
- Modified U-net architecture to predict cell centroids & radial components



UNet architecture

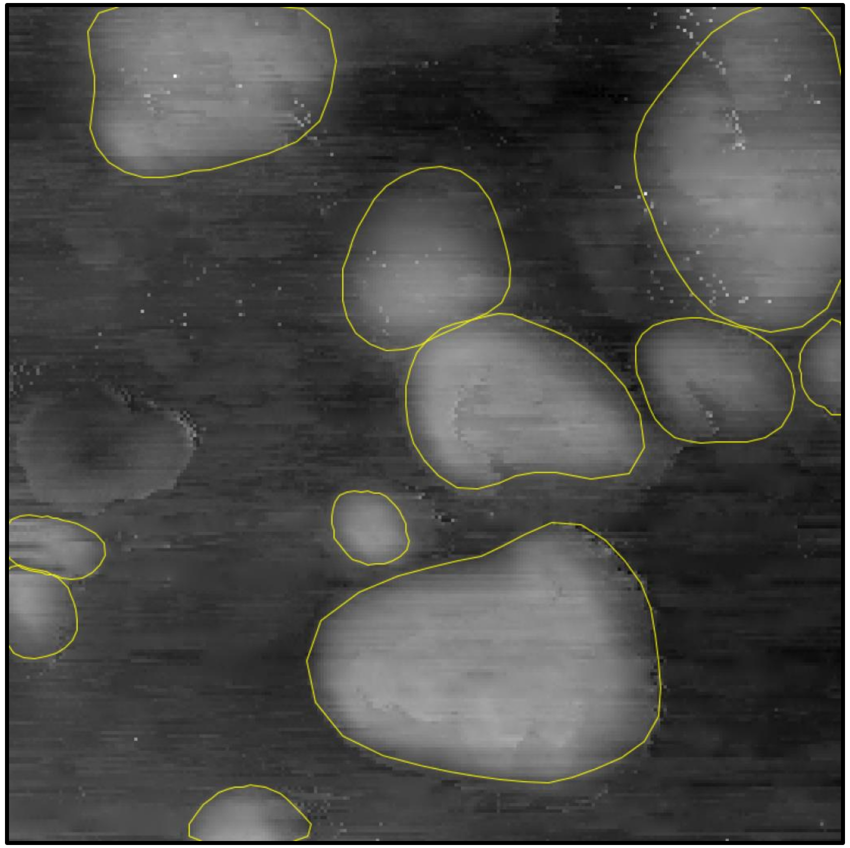
Cell centroid map ($h \times w \times 1$)

Radial components map ($h \times w \times n$)

StarDist

Stardist

- Want to identify *individual* blisters accurately (instance segmentation)
- Developed to provide better instance segmentation in biological images
- Great for segmenting star-convex polygons
- Modified U-net architecture to predict cell centroids & radial components
- Thus, produces coordinate data for each blister



11 blisters → 11 sets of coordinates

Name	Date Modified	Size	Kind
0026-0062.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0049-0224.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0077-0128.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0110-0249.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0114-0217.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0121-0159.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0160-0111.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0165-0014.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0185-0010.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0198-0146.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document
0247-0074.roi	Jul 17, 2024 at 11:20 AM	530 bytes	Document