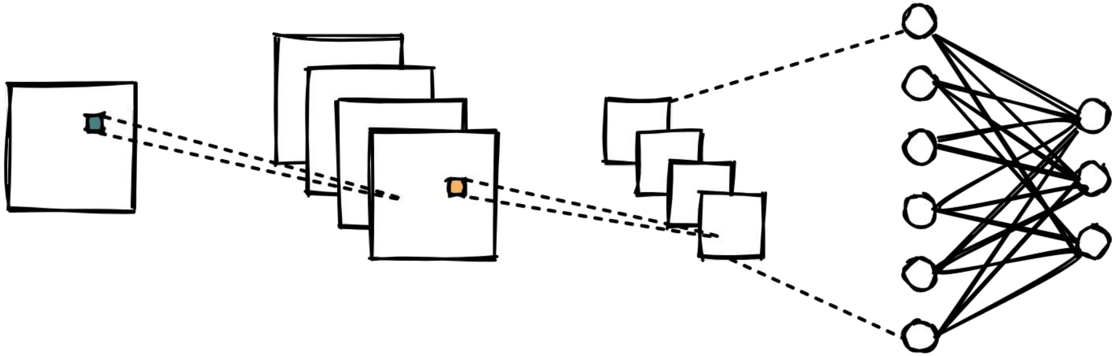
Neural Networks

Part 3



Week 17

Middlesex University Dubai; CST4050 Fall21;
Instructor: Dr. Ivan Reznikov

Plan

- Unstable gradients:
  - Vanishing
  - Exploding
- Computer vision
- Convolutional Neural Networks (CNN)
  - Convolution
  - Pooling
  - Transfer learning

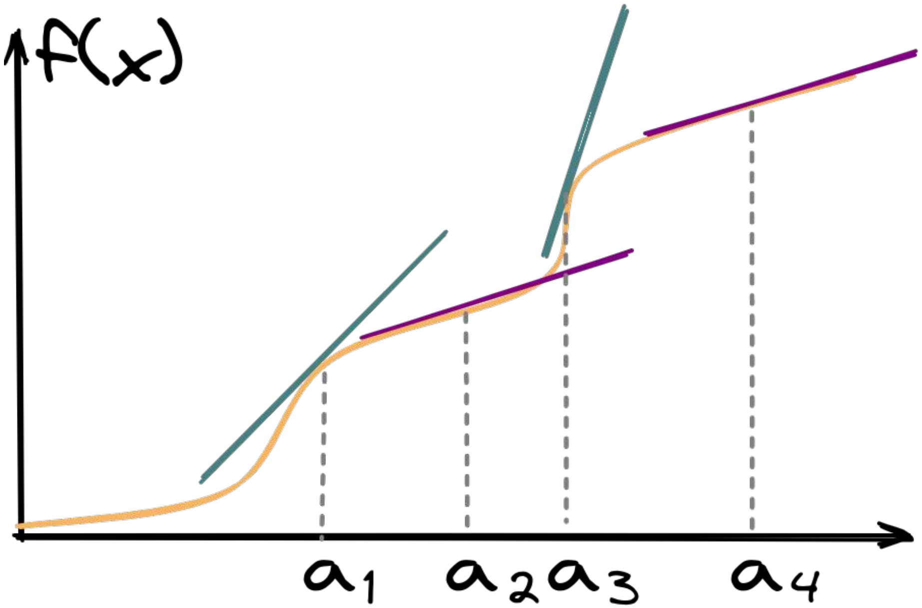
Reminder: Derivatives

Derivative is a measure of the rate at which the value **f(x)** changes with respect to the change of the variable a.

The derivative of the function **f(x)**, evaluated at **x=a** gives the slope of the curve at **x=a**.

Speed is the derivative of distance with respect to time:

$$\text{speed} = \frac{d(\text{distance})}{d(\text{time})}$$



Reminder: Gradient Descent

Our derivative: $28a - 46.32$

current a value = -0.2568

Step1:

$28 \times (-0.2568) - 46.32 = -53.51$

step_size = $-53.51 \times 0.01 = -0.5351$

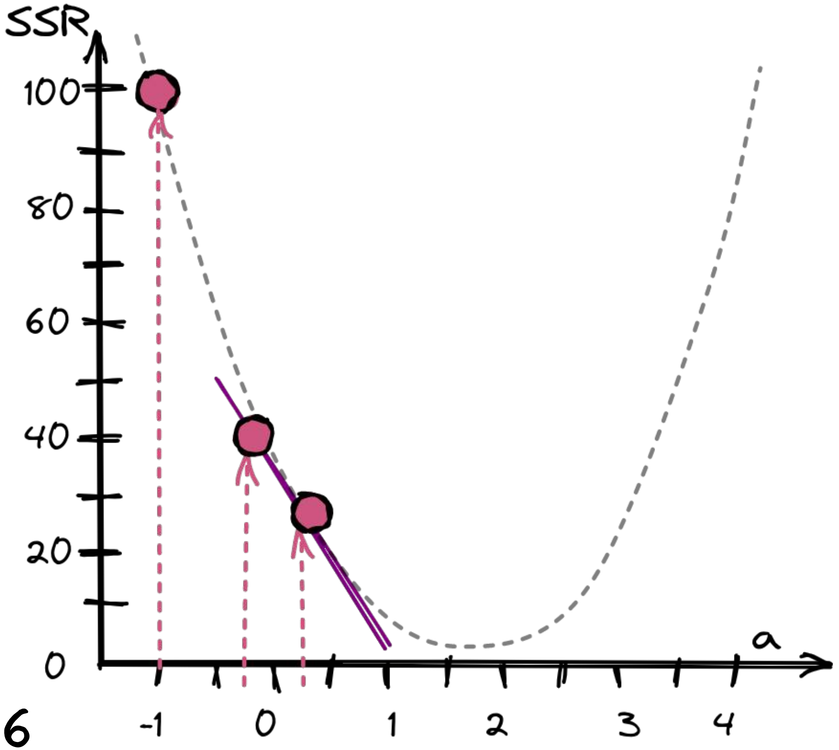
new slope = $-0.2568 - (-0.5351) = 0.2783$

Step2:

$28 \times 0.2783 - 46.32 = -38.53$

step_size = $-38.53 \times 0.01 = -0.3853$

new slope = $0.2783 - (-0.3853) = 0.663576$



Reminder: weights and biases

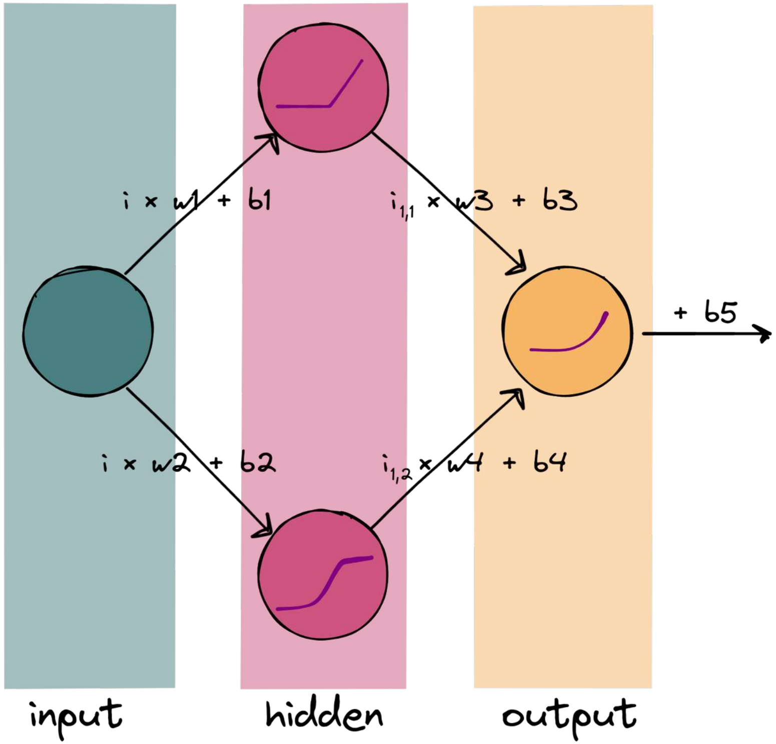
$$\frac{d(SSR)}{d(b5)} = \frac{d(SSR)}{d(Pred)} \times \frac{d(Pred)}{d(b5)}$$

$$\frac{d(SSR)}{d(b3)} = \frac{d(SSR)}{d(Pred)} \times \frac{d(Pred)}{d(b3)}$$

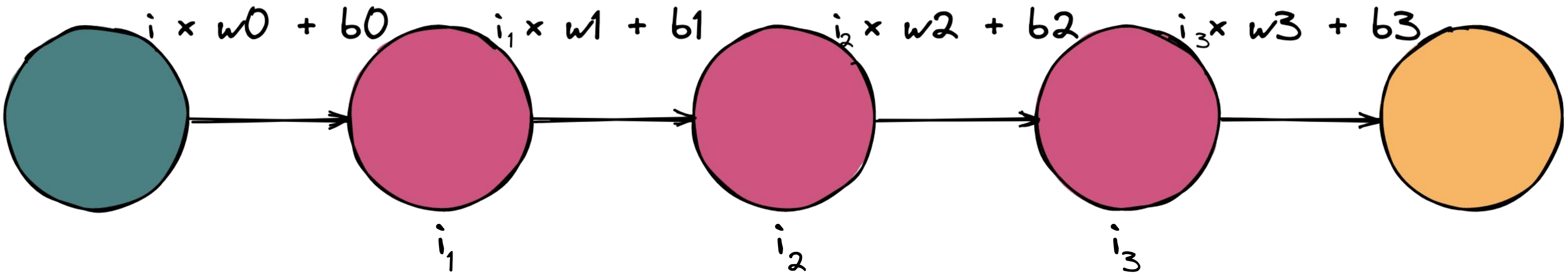
$$\frac{d(SSR)}{d(w3)} = \frac{d(SSR)}{d(Pred)} \times \frac{d(Pred)}{d(w3)}$$

$$\frac{d(SSR)}{d(b1)} = \frac{d(SSR)}{d(Pred)} \times \frac{d(Pred)}{d(i_{1,1})} \times \frac{d(i_{1,1})}{d(b1)}$$

$$\frac{d(SSR)}{d(w1)} = \frac{d(SSR)}{d(Pred)} \times \frac{d(Pred)}{d(i_{1,1})} \times \frac{d(i_{1,1})}{d(w1)}$$



Backpropagation



$$\frac{d(SSR)}{d(w0)} = \frac{d(SSR)}{d(i_3)} \times \frac{d(i_3)}{d(i_2)} \times \frac{d(i_2)}{d(i_1)} \times \frac{d(i_1)}{d(w0)}$$

Backpropagation

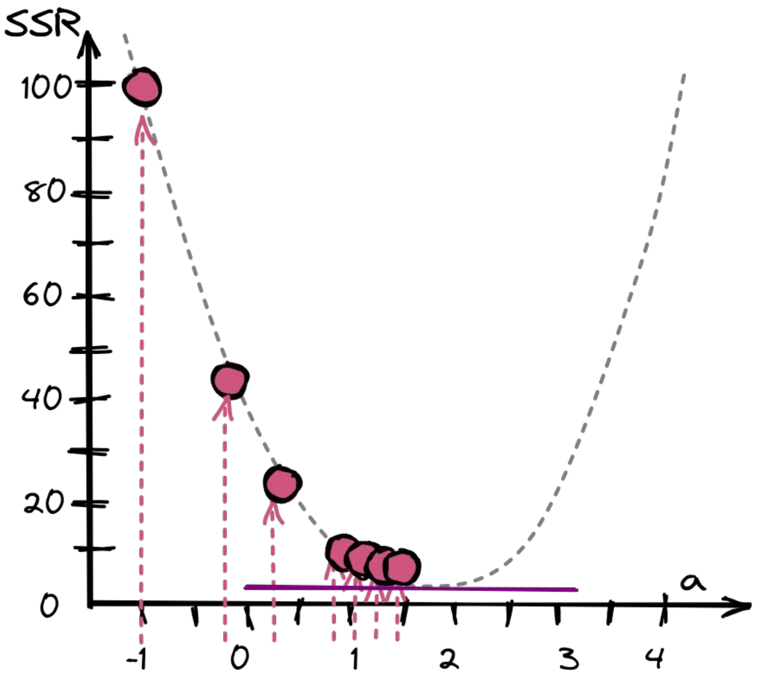
$$\frac{d(SSR)}{d(w0)} = \frac{d(SSR)}{d(i_3)} \times \frac{d(i_3)}{d(i_2)} \times \frac{d(i_2)}{d(i_1)} \times \frac{d(i_1)}{d(w0)}$$

w0 = 10

LR (learning rate) = 0.01

$$\text{step\_size} = \frac{d(SSR)}{d(w0)} \times LR$$

new slope = w0 - step_size



Unstable gradients: vanishing

$$\frac{d(SSR)}{d(w0)} = \frac{d(SSR)}{d(i_3)} \times \frac{d(i_3)}{d(i_2)} \times \frac{d(i_2)}{d(i_1)} \times \frac{d(i_1)}{d(w0)}$$

$w0 = 10$

LR (learning rate) = 0.01

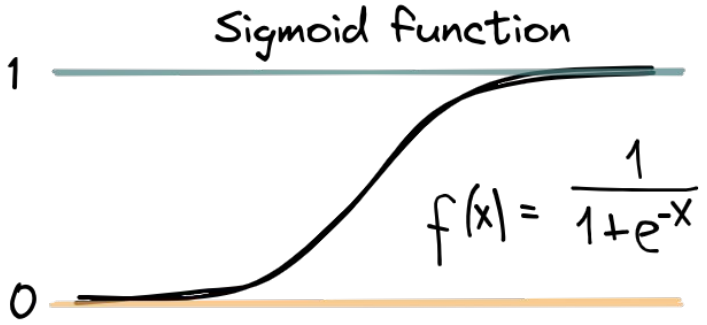
$$\text{step\_size} = \frac{d(SSR)}{d(w0)} \times LR$$

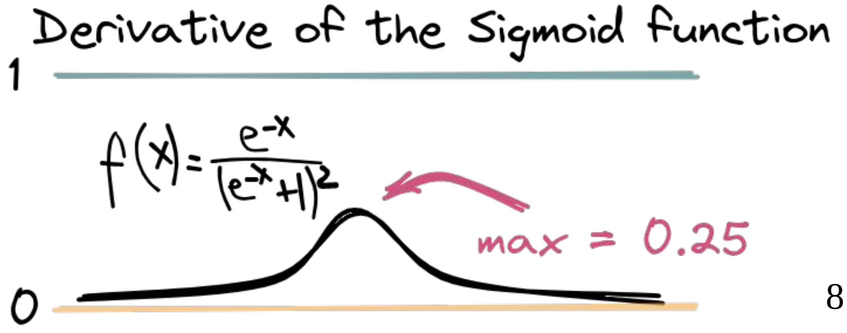
new_slope = w0 − step_size

Example: step_size → 0;

step_size = 0.00005 =>

=> new_slope = 9.99995





Unstable gradients: exploding

$$\frac{d(SSR)}{d(w0)} = \frac{d(SSR)}{d(i_3)} \times \frac{d(i_3)}{d(i_2)} \times \frac{d(i_2)}{d(i_1)} \times \frac{d(i_1)}{d(w0)}$$

$w0 = 10$

LR (learning rate) = 0.1

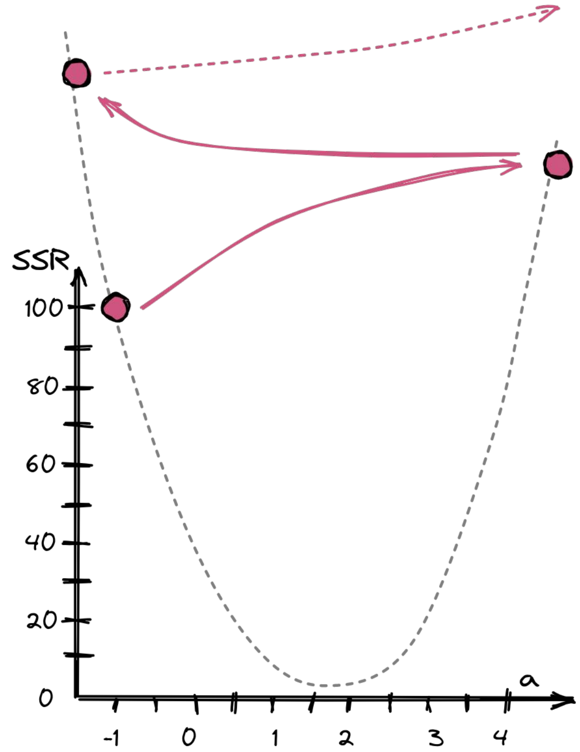
$$\text{step\_size} = \frac{d(SSR)}{d(w0)} \times LR$$

new_slope = w0 − step_size

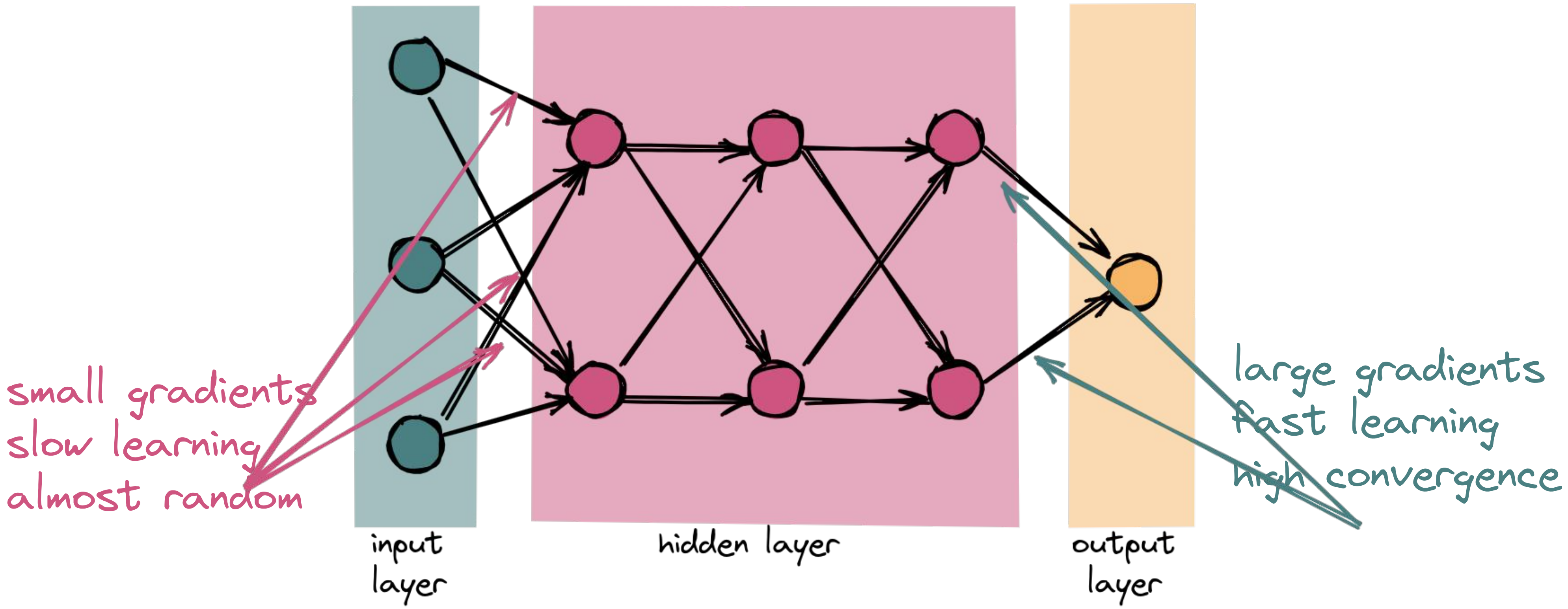
Example: step_size → 5;

step_size = 62.5 =>

=> new_slope = 10-(-62.5) = -52.5



Unstable gradients: vanishing



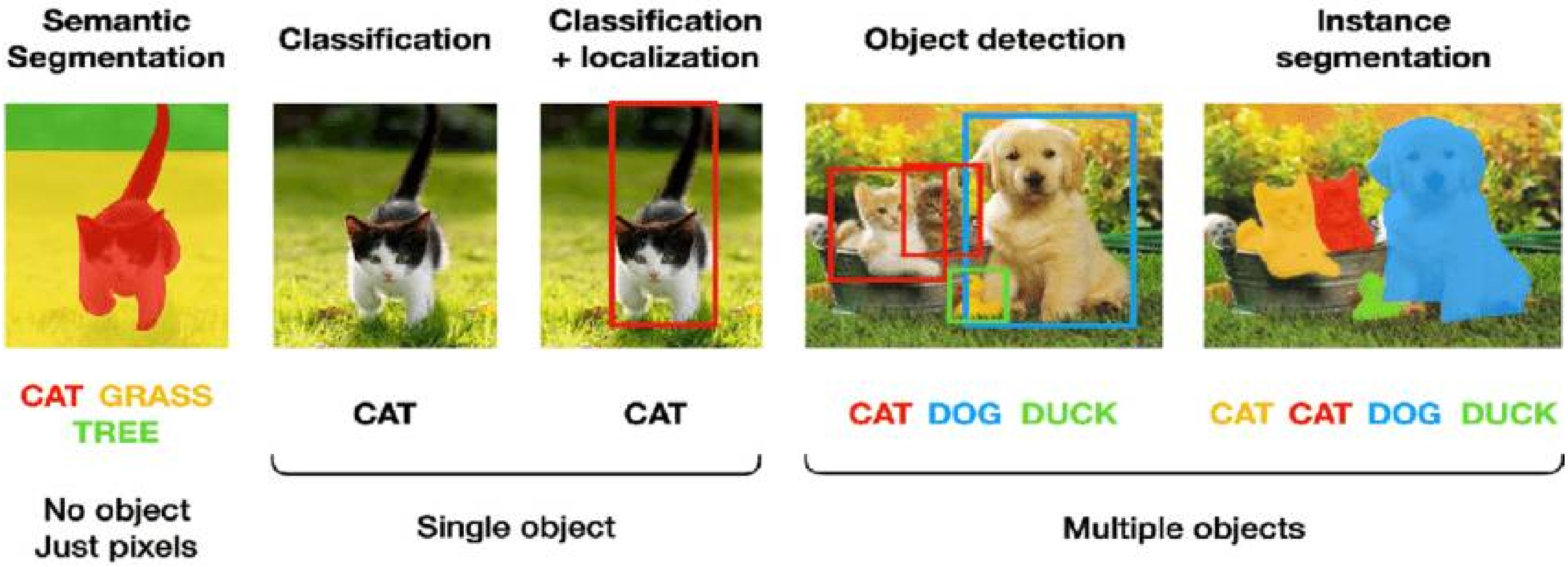
Nowadays solution:

The unstable gradient problem is currently already solved and can be managed with the following improvements:

- better starting weights initialization
- activation functions (ReLU (gradient is 0 or 1), etc)
- regularization techniques:
    - dropout
    - batch normalization
    - data augmentation
- robust gradient descent optimizer

Computer vision

Computer Vision (CV) is a field of study that helps computers “see” and understand the content of digital images and videos.



Convolutional Neural Networks

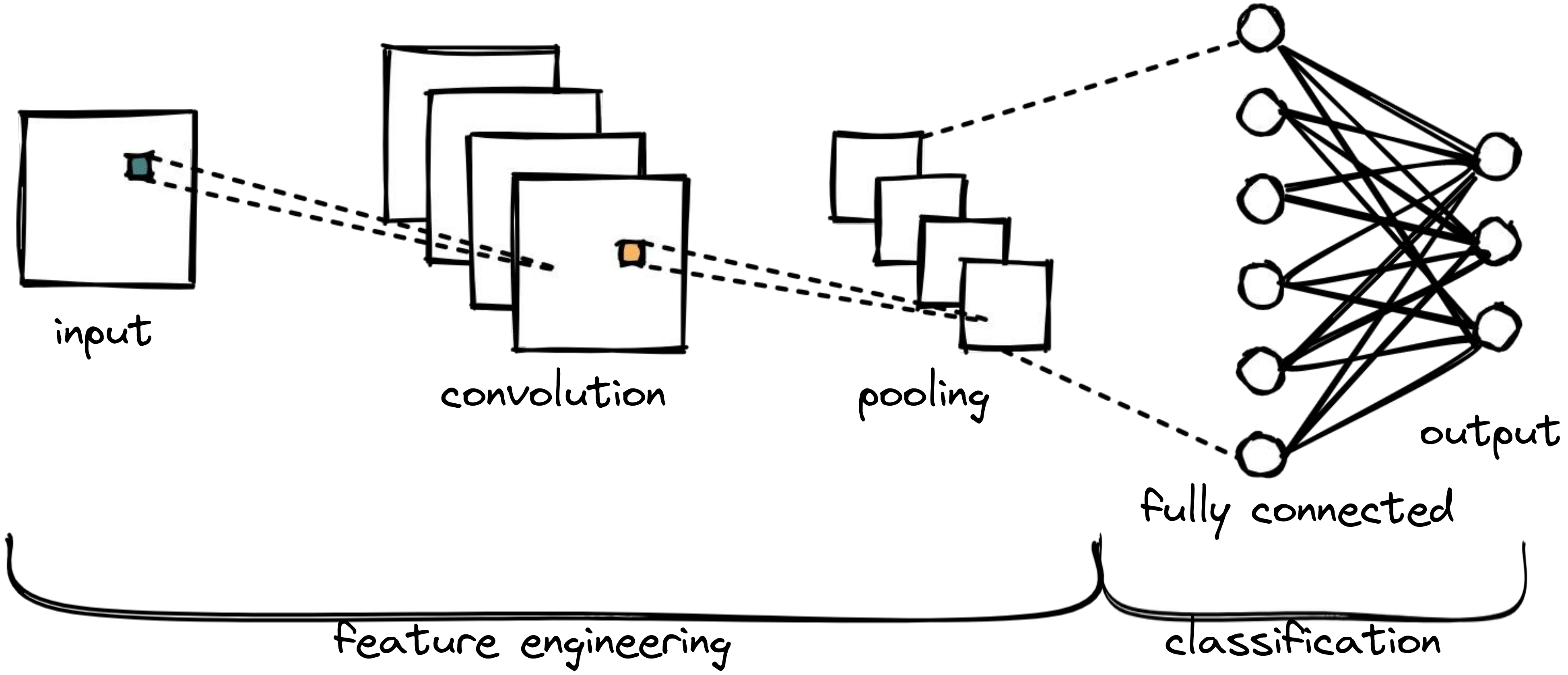
A convolutional neural network (CNN) is a network architecture for deep learning that learns directly from data, eliminating manual feature extraction.

CNNs are especially useful for finding patterns in pictures to recognize objects, faces, and scenes.

Most popular applications:

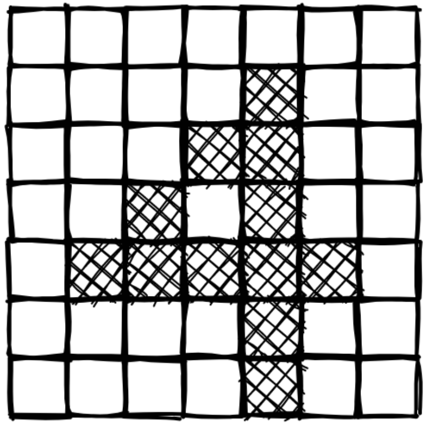
- Face detection
- Facial emotion recognition
- Object detection
- Self-driving or autonomous cars
- Image captioning
- Biometric authentication
- Auto translation
- etc.

Convolutional Neural Networks



Great demo: https://poloclub.github.io/cnn-explainer/

CNN: Case1



Our target image.

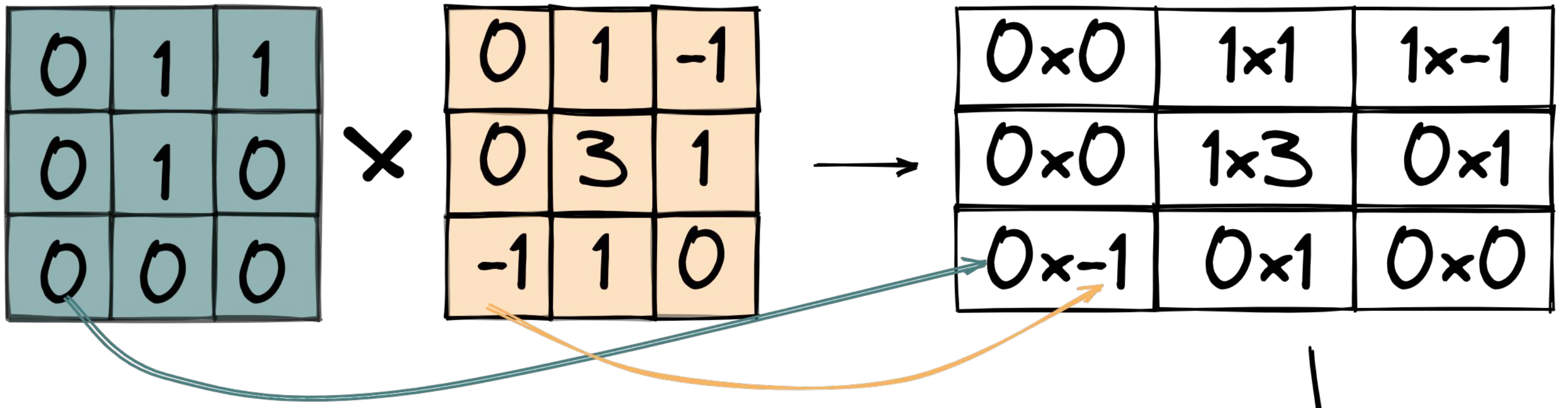
We'll convert it into a 2D array with 0s and 1s

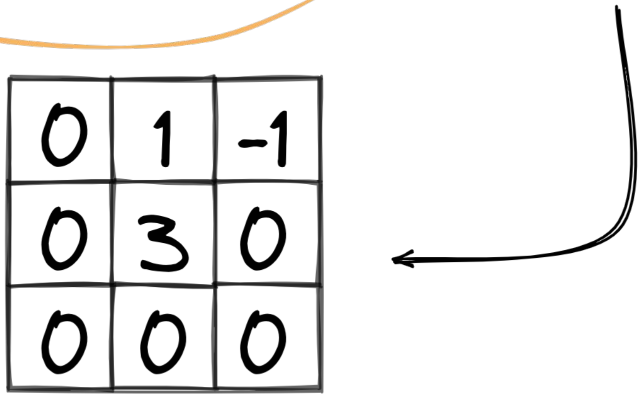
0	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	1	1	1	1	1	0
0	0	0	0	1	0	0
0	0	0	0	1	0	0

Our convolution matrix. Consider it as "weights" for some neuron

0	1	-1
0	3	1
-1	1	0

Convolution multiplication



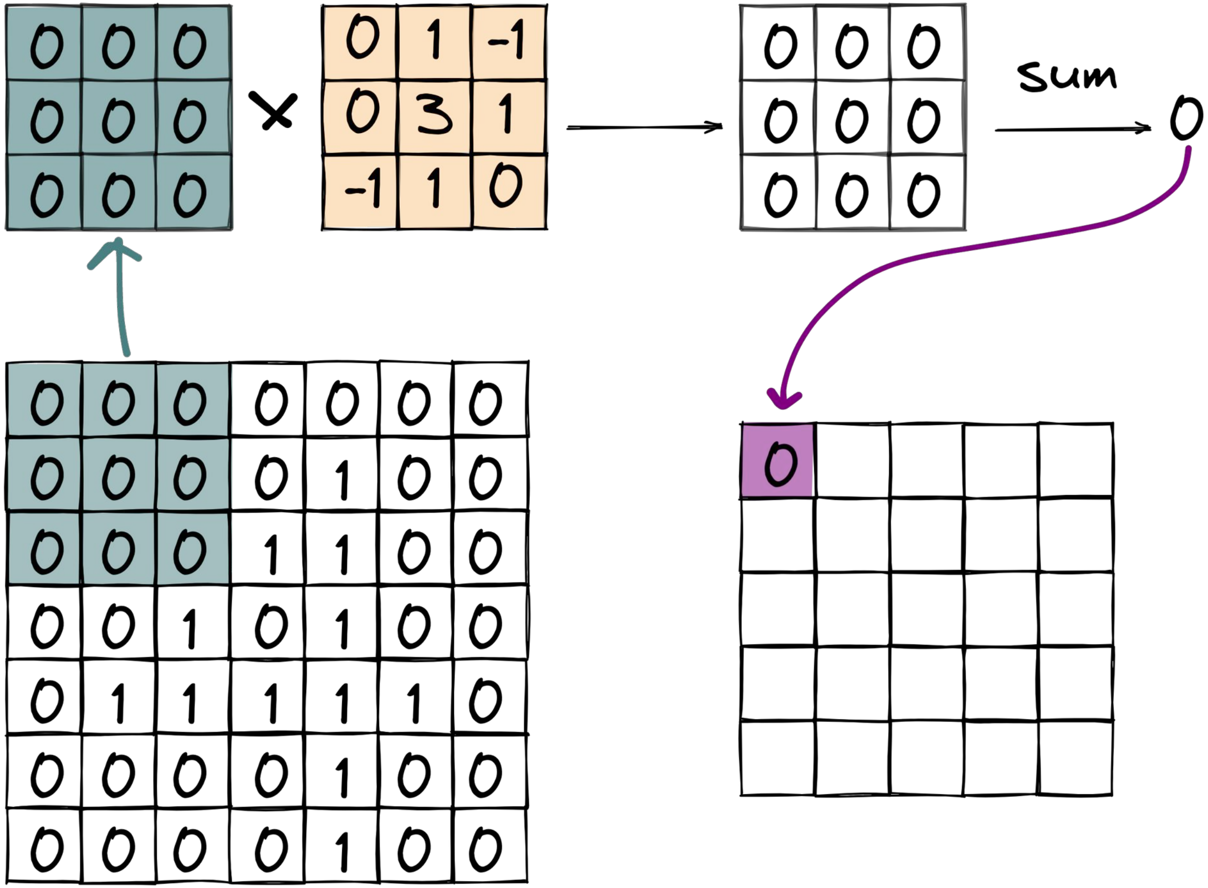


Unlike regular matrix multiplication, we're multiplying elements of same row-column indexes

Convolution: Step1.1

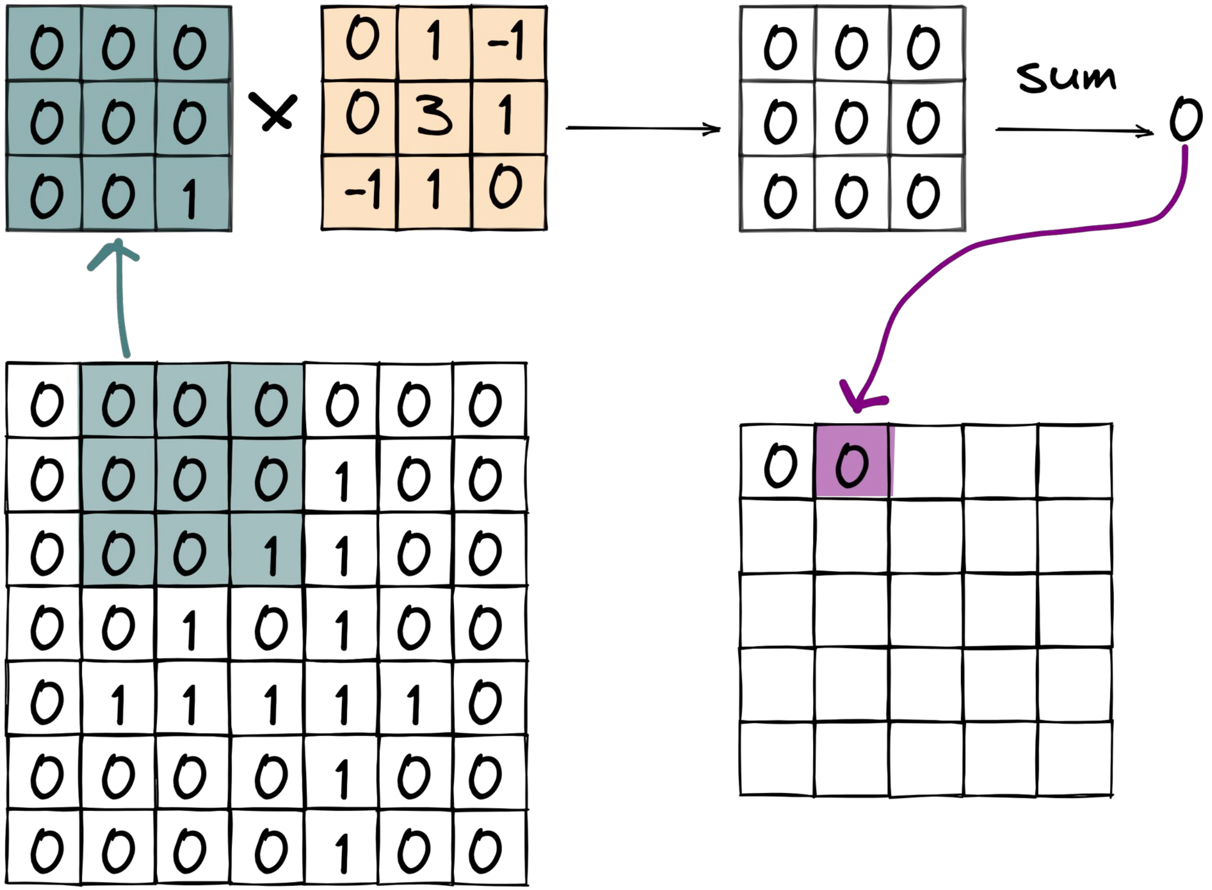
The idea of convolution is to study groups of several pixels rather than a single pixel, as it might give us more information.

We're starting with the left top corner. As a result, we store the total sum of the multiplication matrix.



Convolution: Step1.2

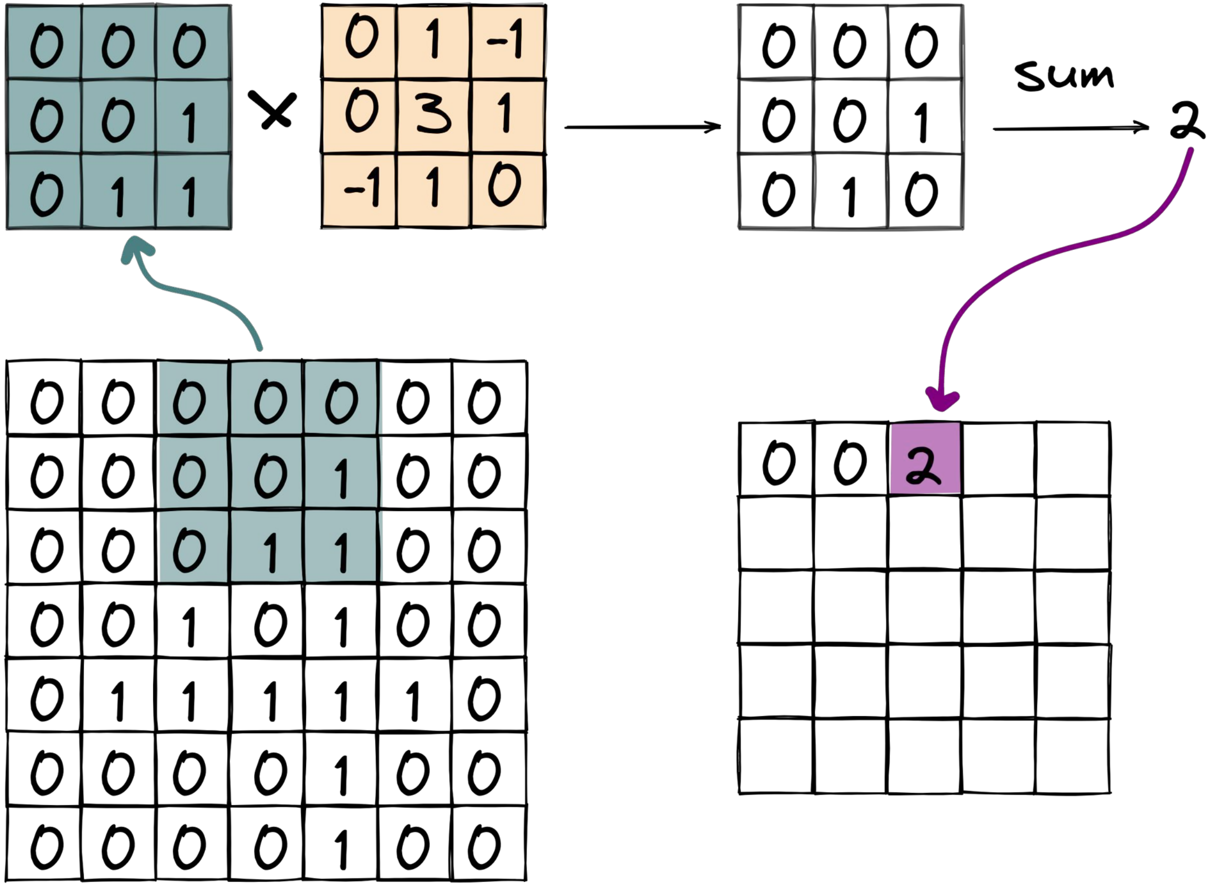
Next step, we'll slide our convolution window to the right. We'll calculate the resulting matrix's sum and store it in the new matrix.



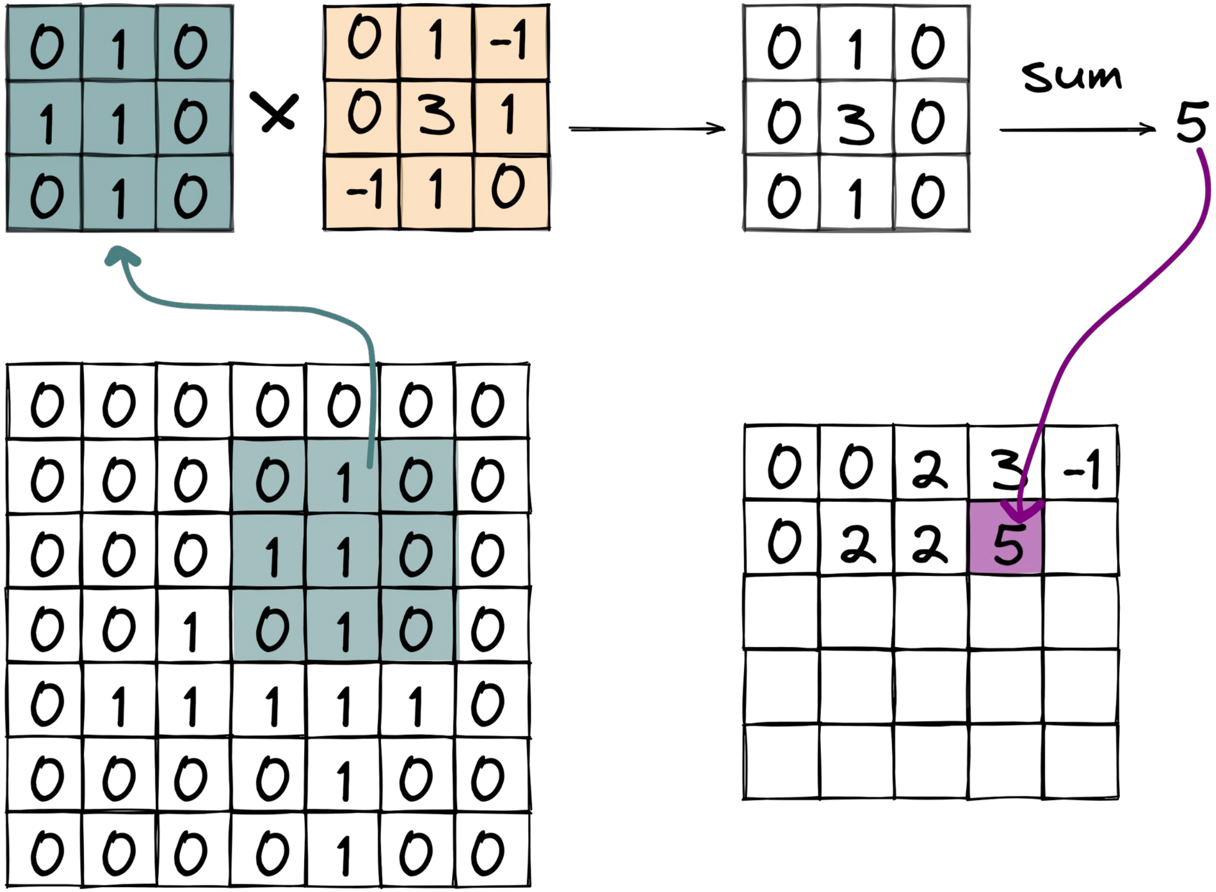
Convolution: Step1.3

We'll perform the same operation again. This time the sum isn't 0, and we write the resulted sum (2) in our result matrix.

We'll perform the same operation till we reach the right border. We'll slide 1 row down and start over.



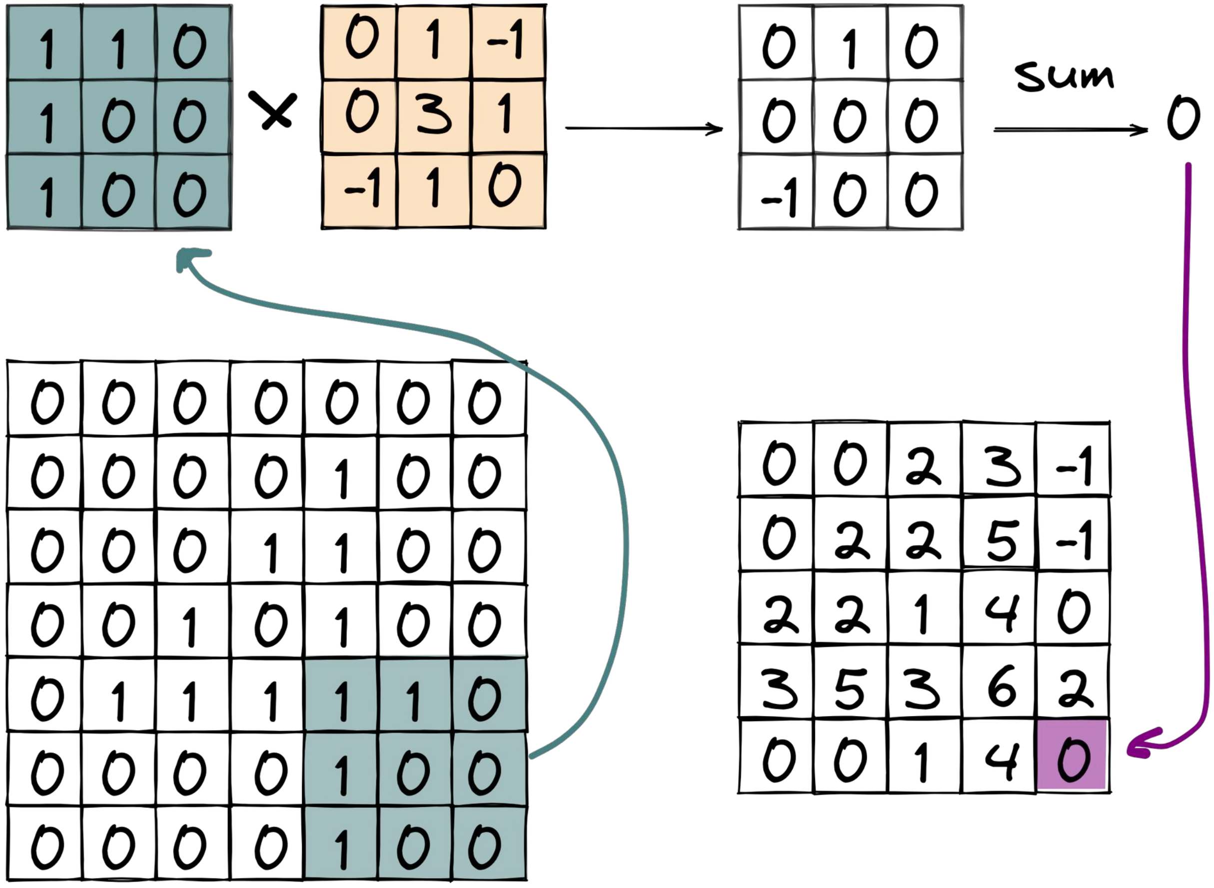
Convolution: Step1.K



Convolution: Step1.Z

Now we've reached the bottom right corner. Our result matrix is filled.

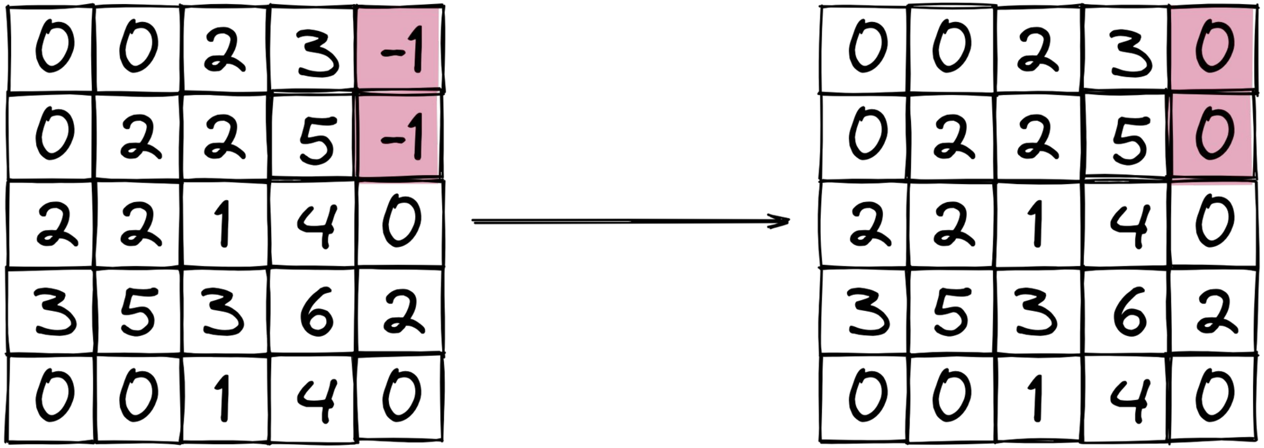
Our next step is usually activation.



CNN: Step2. ReLU activation

In this example, we'll use ReLU activation, a popular choice for CNNs.

In our example, one can notice that negative top right numbers were replaced with 0s.



CNN: Result of ReLU activation

The result matrix after activation is present on the right.

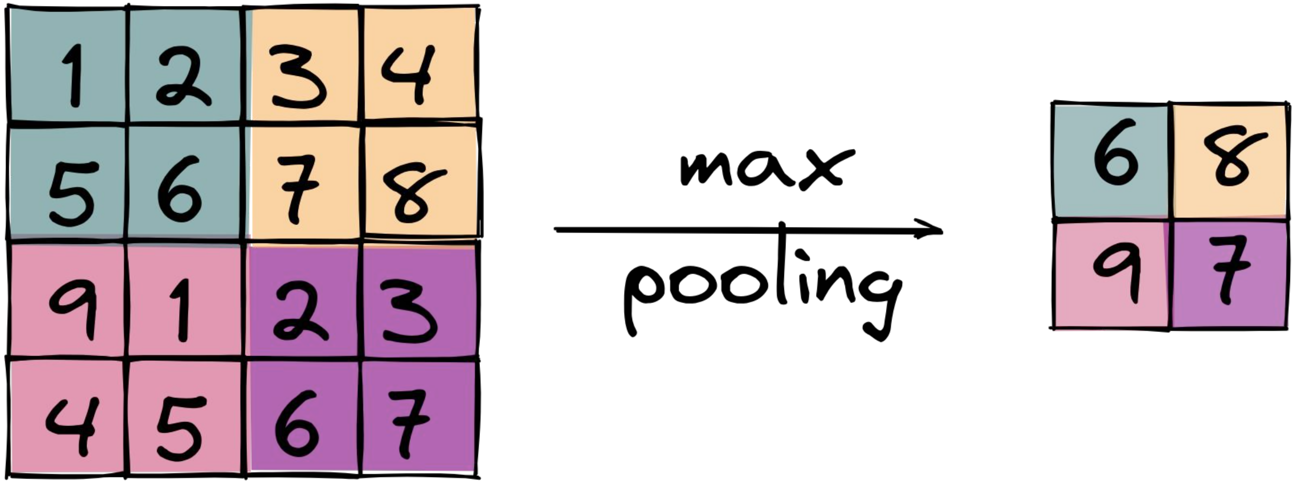
Notice that we've decreased the number of elements without much loss in quality.

Usually, several convolution operations take place in real-life CNNs.

0	0	2	3	0
0	2	2	5	0
2	2	1	4	0
3	5	3	6	2
0	0	1	4	0

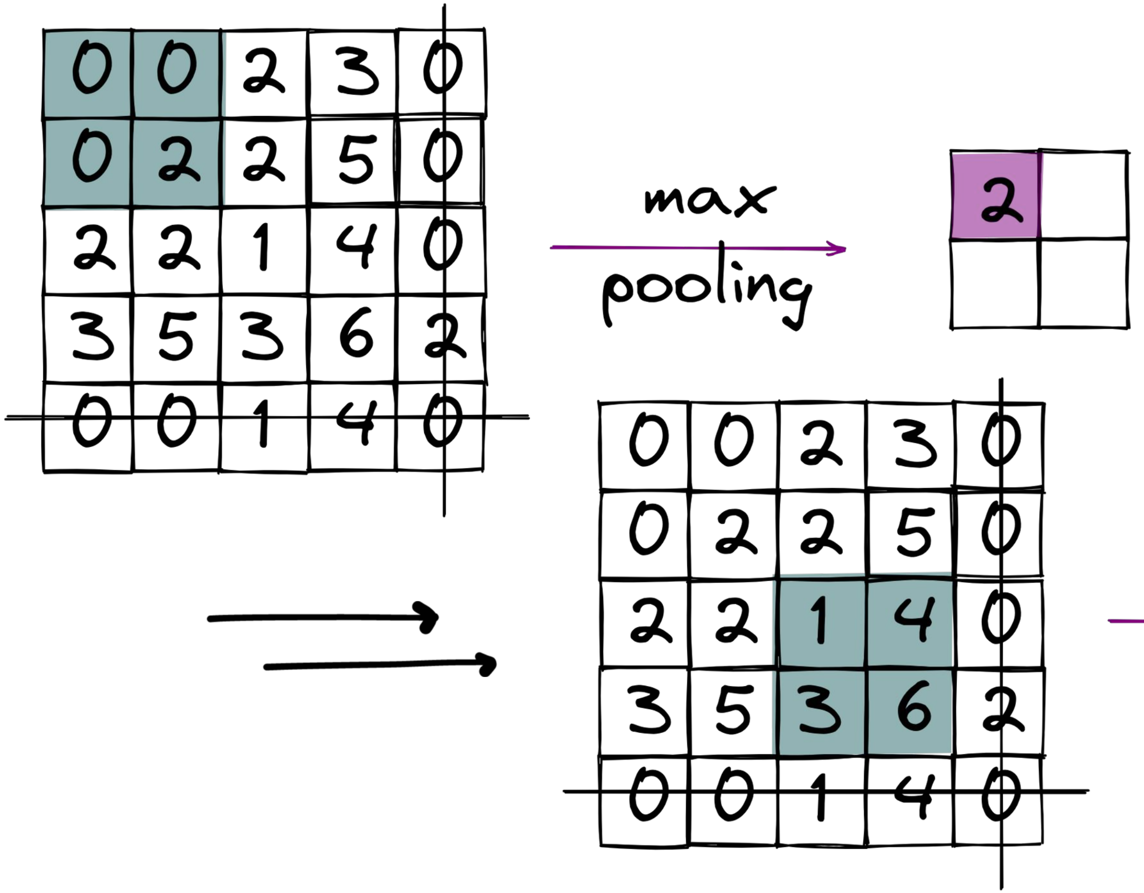
Max pooling concept

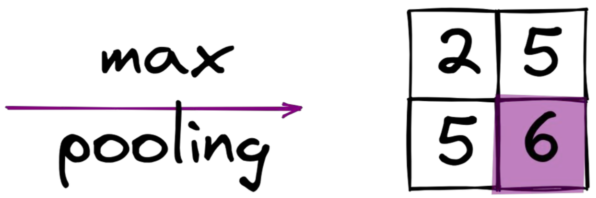
Pooling is another quite often operation performed in CNNs. Usually, a square matrix 2×2 is used, and the maximum value is taken, as it is shown on the right.



One may say that convolution is looking for local features, whereas pooling is responsible for more "global" ones. Imagine it as first detecting edges in some image, that would be used to form facial features later. This is how simplest facial recognition might work.

CNN: Max pooling



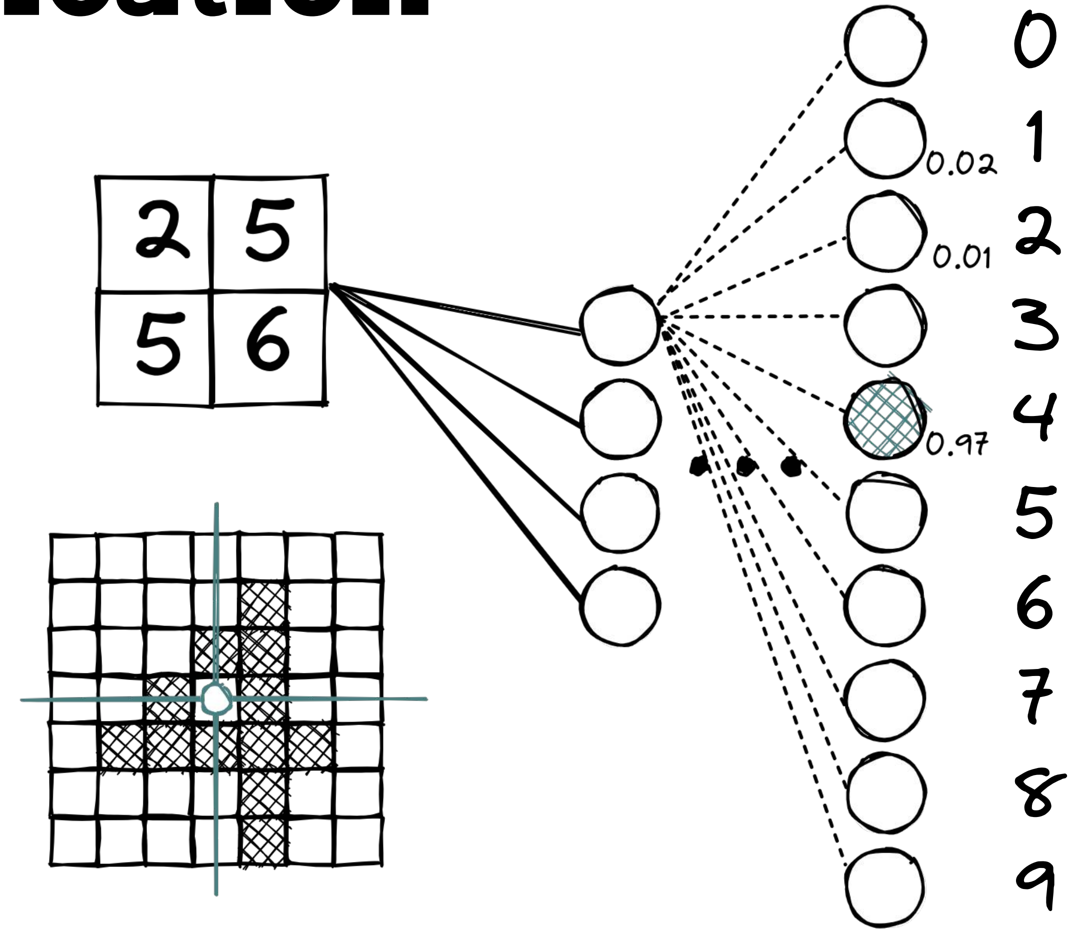


Our matrix is 5×5, whereas the pooling matrix is 2×2. In cases where we have extra rows and columns, one of the strategies is to exclude them.
Our result matrix is 2×2.

CNN: Classification

If we compare our initial image with the resulting matrix, we might notice that it almost replicates the number of pixels in the respected squares. Of course, this is a simple example, but you may see how effective our CNN is.

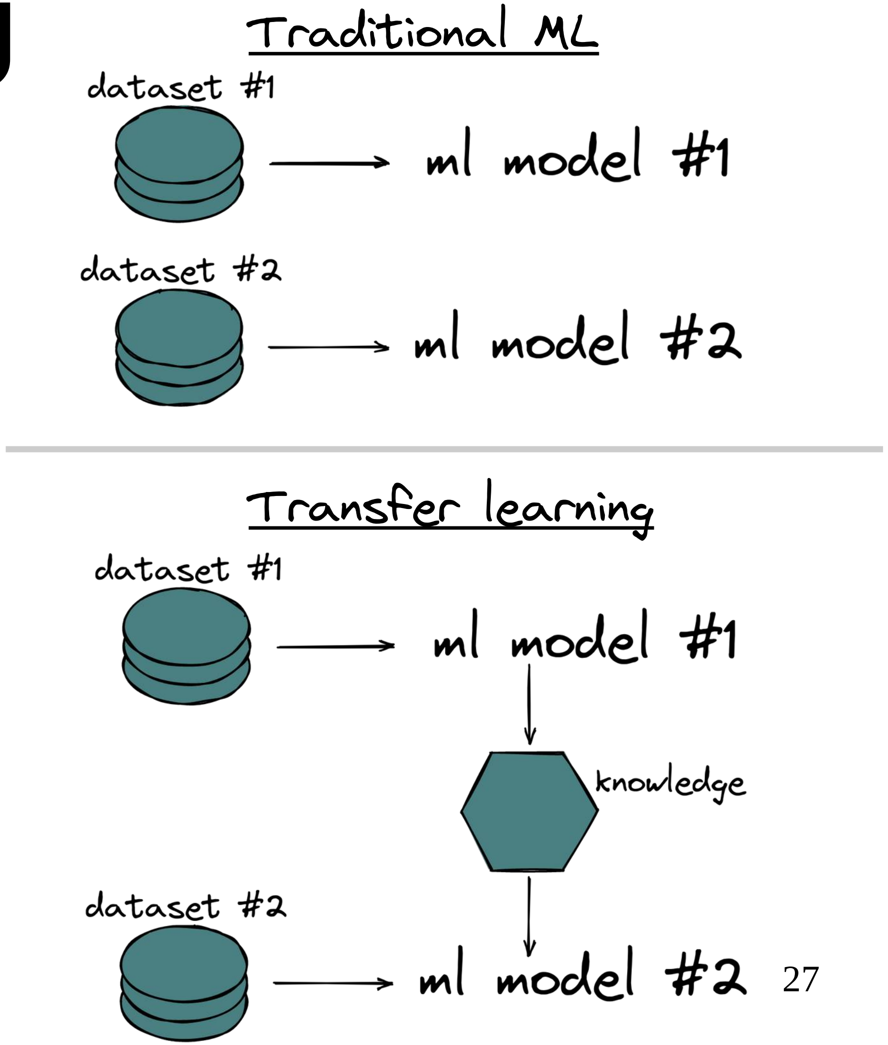
Our next step will be to flatten the 2×2 matrix into 4 nodes. From this point on, we may use any neural network strategy we feel is needed.



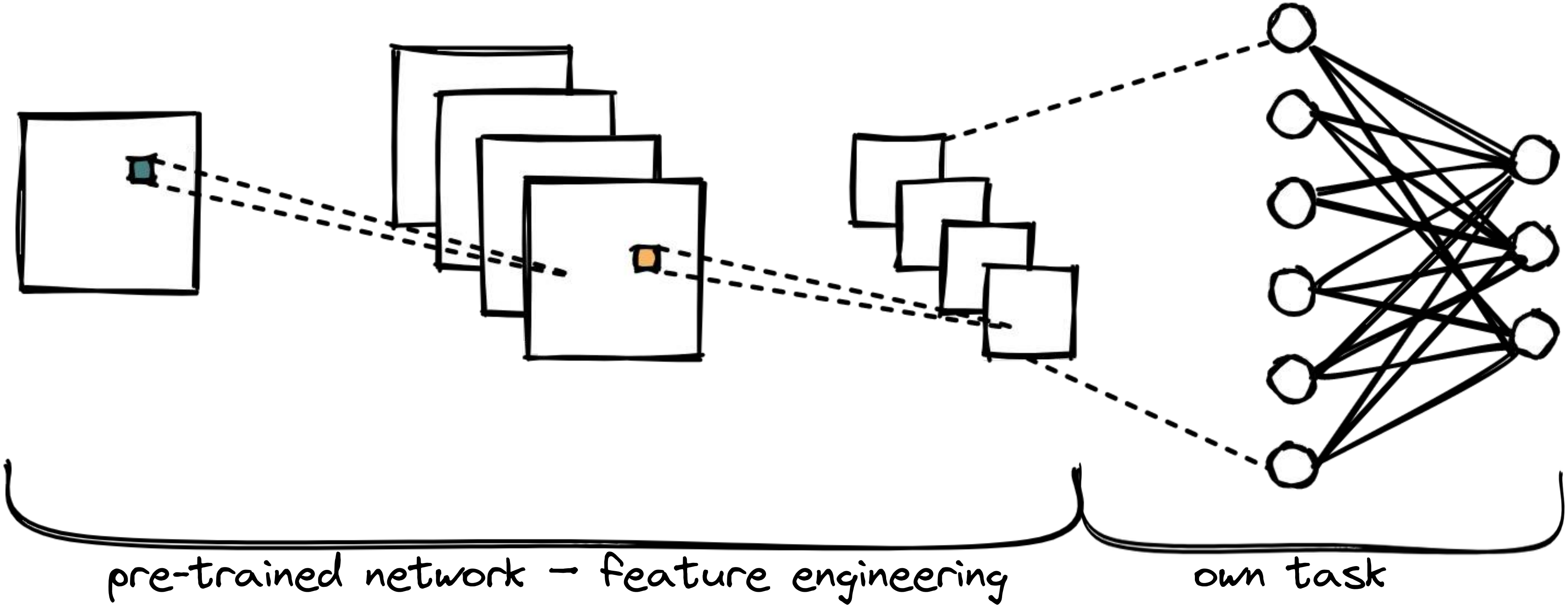
Transfer learning

Transfer learning (TL) is a technique that concentrates on storing knowledge gained while solving one task and applying it to a different but related problem.

We can use "pre-trained" model outputs as inputs in our neural network. In such a case, we're running a pre-trained model similar to code functions.



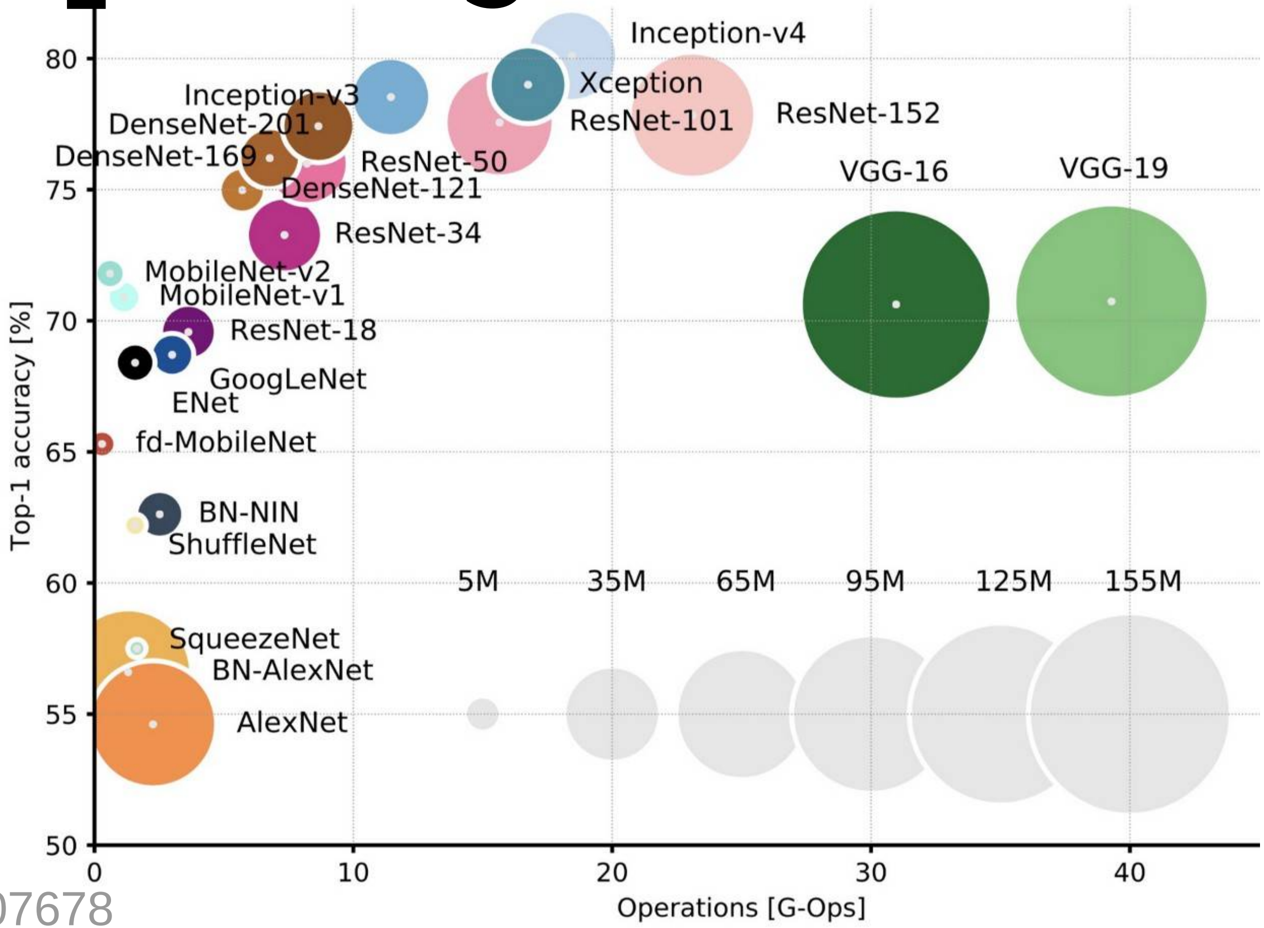
CNN: Pre-trained networks



https://huggingface.co/deepmind/vision-perceiver-conv
https://huggingface.co/facebook/detr-resnet-50

CNN: Max pooling

Some of the networks shown are learned on millions of images. Using transfer learning, we can benefit from using top models for our needs.

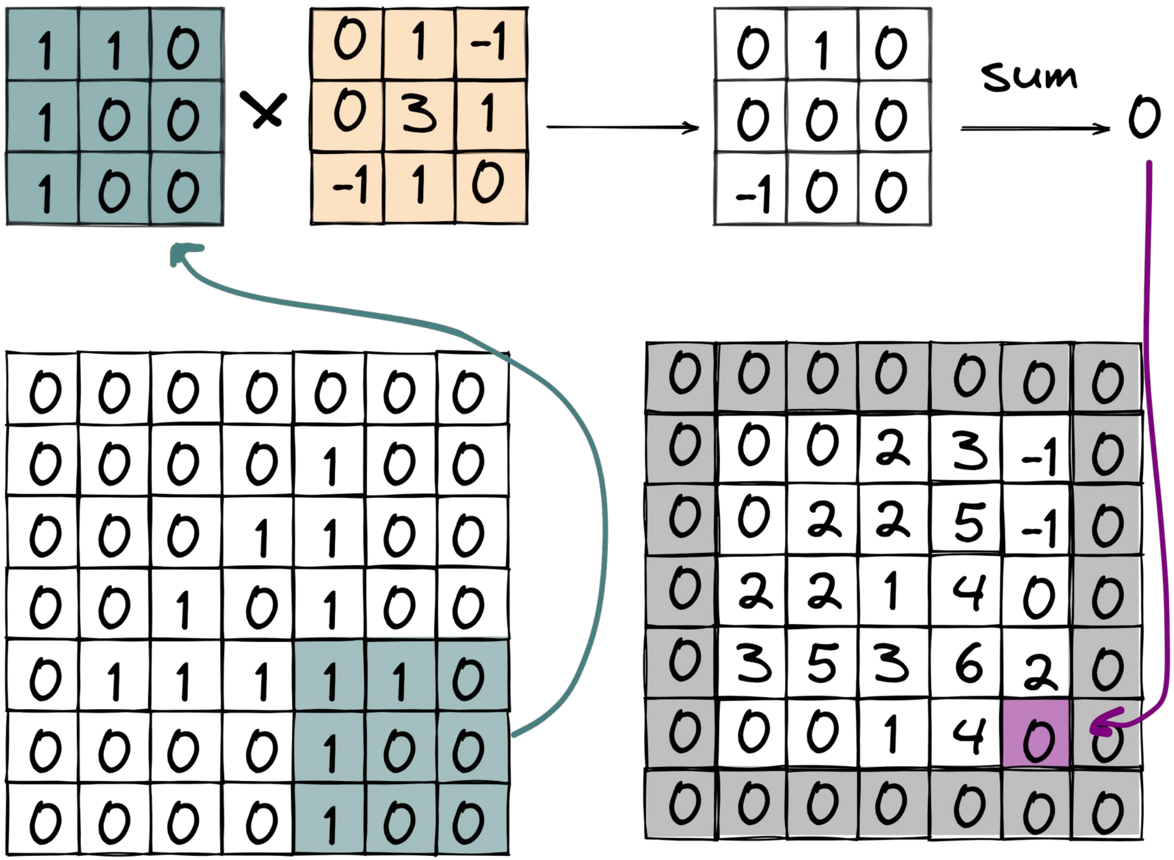


https://arxiv.org/abs/1605.07678

Convolution: Practical tips

Zero padding:

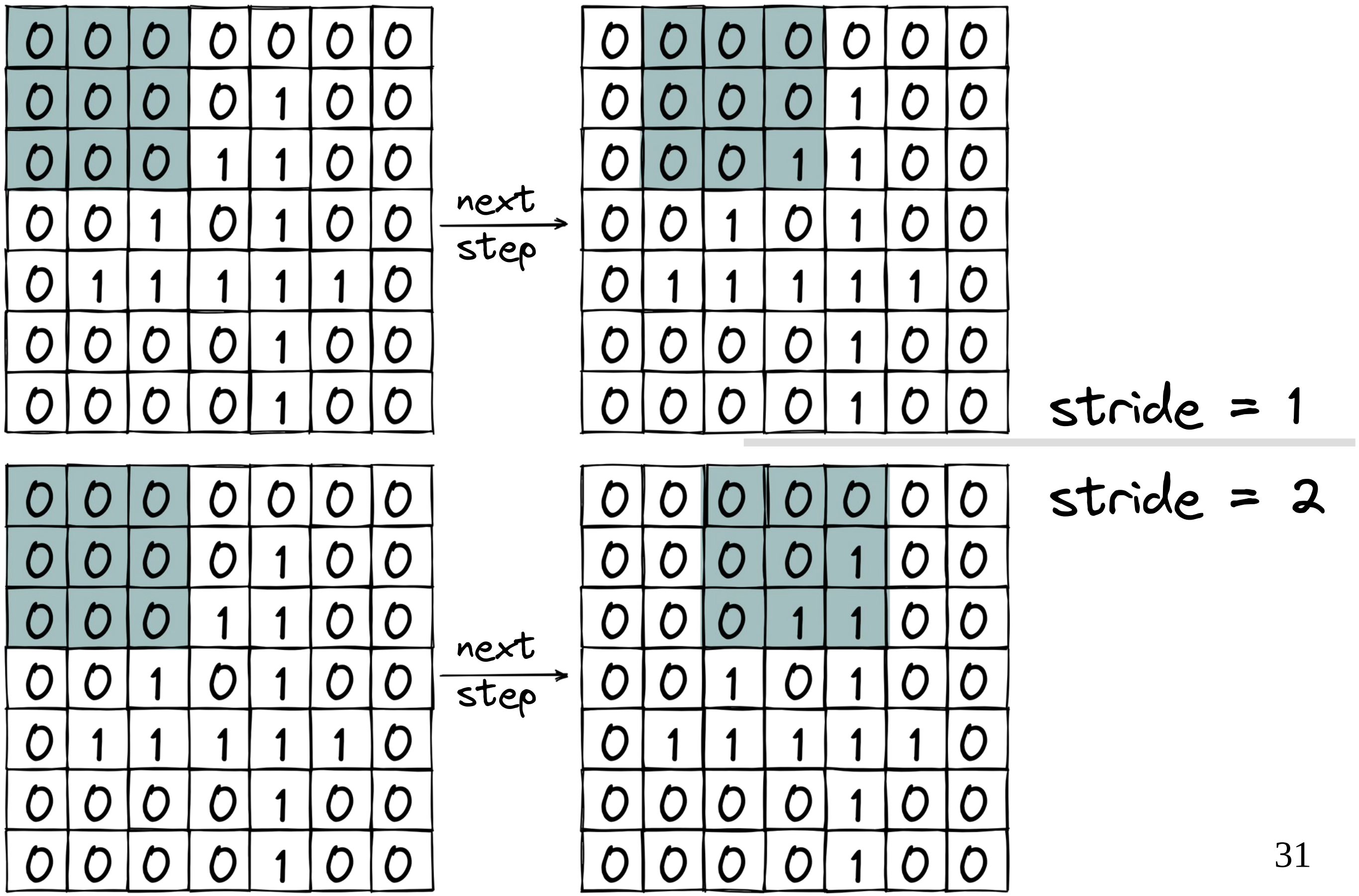
Preserves size, as input size = output size



Convolution: Practical tips

Stride:

The further apart pixels are from each other, the less they are correlated. It May be used for downsampling the input



Other implementations

Text summarization:

The tower is 324 metres (1,063 ft) tall, about the same height as an 81-storey building, and the tallest structure in Paris. Its base is square, measuring 125 metres (410 ft) on each side. During its construction, the Eiffel Tower surpassed the Washington Monument to become the tallest man-made structure in the world, a title it held for 41 years until the Chrysler Building in New York City was finished in 1930. It was the first structure to reach a height of 300 metres. Due to the addition of a broadcasting aerial at the top of the tower in 1957, it is now taller than the Chrysler Building by 5.2 metres (17 ft). Excluding transmitters, the Eiffel Tower is the second tallest free-standing structure in France after the Millau Viaduct.

<u>Summary:</u>

The tower is 324 metres (1,063 ft) tall, about the same height as an 81-storey building. Its base is square, measuring 125 metres (410 ft) on each side. During its construction, the Eiffel Tower surpassed the Washington Monument to become the tallest man-made structure in the world.

https://huggingface.co/facebook/bart-large-cnn