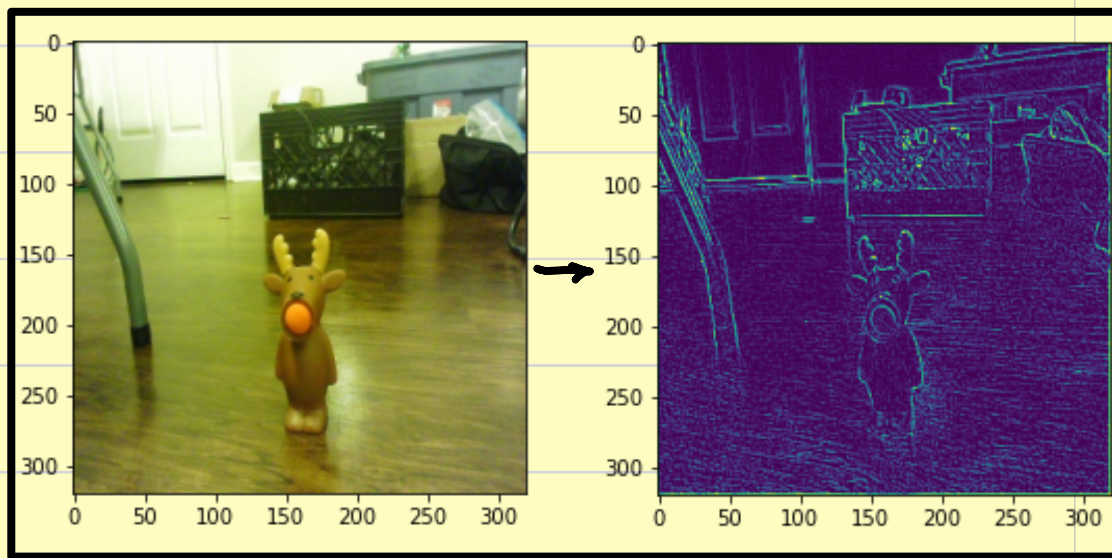
20251030_Business_Intelligence_WIN6

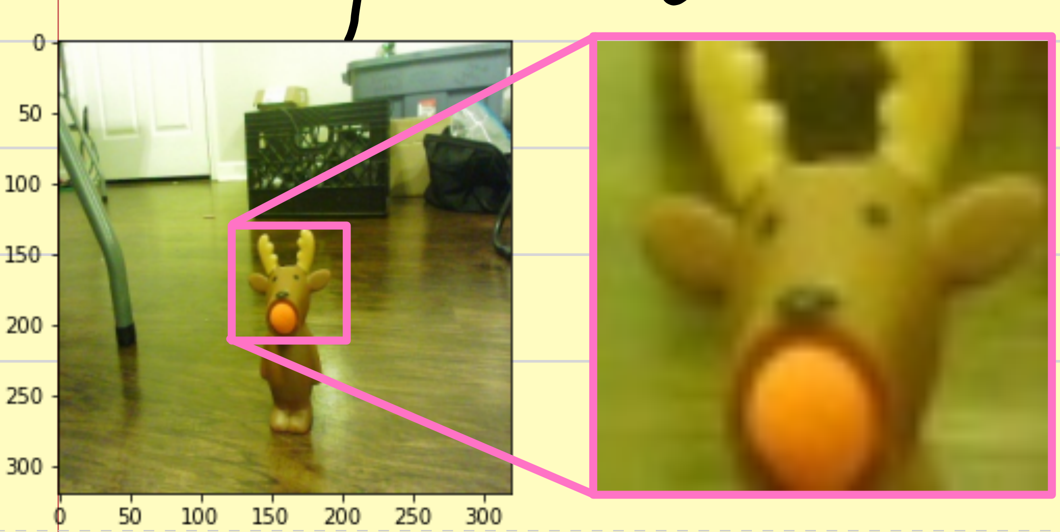
Deep Learning by hand (II)

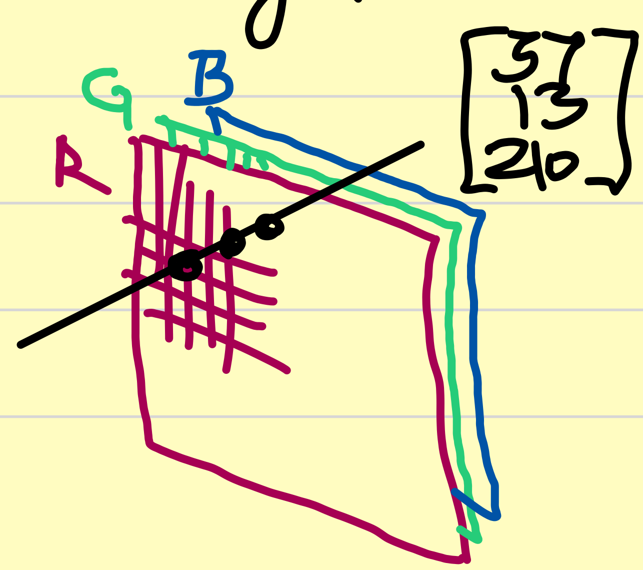
„Convolution"



- Convolution is a technique used in image processing. Is the process of summing and multiplying each element of the input image with its local neighbours.
- The convolution uses FILTERS to extract the relevant information we are interested in. These filters are called KERNELS
- A kernel is in this case a matrix that shapes the weights of the convolution.
- $(f \circ g)(x) = f(g(x))$

Image processing. What is an image?





- Each pixel has 3 channels: Red, Green, Blue.
- Each value can be between [0, 255].
- Each pixel is represented by 3 values: [37, 13, 210]

256 values: 1 byte = 8 bits $\quad 2^8 = 256$

- Each bit can be either 0 or 1.

Now we define some Kernels...

- Kernels can have many shapes & sizes.
- Kernels can be used for attention, blurring, edgerecognition, ...
- Edge Recognition:

1	0	-1
0	0	0
-1	0	1

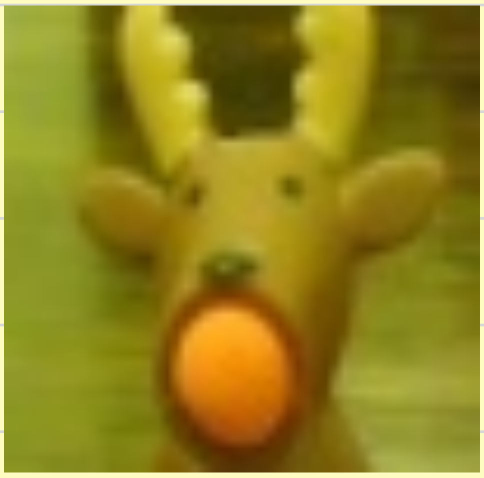
Only the edges are non-zero.

- Attention Kernel:

0	-1	0
-1	4	-1
0	-1	0

The middle is highlighted (x4).

Example:



$$
\rightarrow
\begin{array}{ccc}
R & G & B \\
\begin{bmatrix} 1 & 1 & 0 & 2 & 1 \\ 1 & \boxed{2} & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 1 & 2 & 2 & 1 & 0 \end{bmatrix} & &
\end{array}
$$

Attention Kernel

0	-1	0
-1	4	-1
0	-1	0

$(f \circ g)(x)$

Convolution : Stride = 1.

$$
\begin{matrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{matrix} \circ \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \longrightarrow \begin{matrix} 1.0 + 1(-1) + 0.0 + \\ 1(-1) + 2.4 + 1.(-1) + \\ 0.0 + 1.(-1) + 2.0 \end{matrix} = 4
$$

$$
\begin{matrix} 1 & 0 & 2 \\ 2 & 1 & 1 \\ 1 & 2 & 1 \end{matrix} \circ \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \longrightarrow \begin{matrix} 1.0 + 0(-1) + 2.0 + \\ 2(-1) + 4.1 + 1(-1) + \\ 1.0 + 2(-1) + 1.0 \end{matrix} = -1
$$

$$
\begin{matrix} 0 & 2 & 1 \\ 1 & 1 & 0 \\ 2 & 1 & 1 \end{matrix} \circ \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \longrightarrow \begin{matrix} 0.0 + 2(-1) + 1.0 + \\ 1(-1) + 4.1 + 0(-1) + \\ 2.0 + 1.(-1) + 1.0 \end{matrix} = 0
$$

$$
\begin{matrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 1 & 2 & 2 \end{matrix} \circ \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \longrightarrow \begin{matrix} 1.0 + 2(-1) + 1.0 + \\ 0.(-1) + 4.1 + 2.(-1) \\ 1.0 + 2(-1) + 2.0 \end{matrix} = -2
$$

Convoluted Red Layer

4	-1	0
-2	…	…

. . .

Padding. in order not to lose Information on the edges of the Image, we can pad the image with zeros.

padding = p = 1

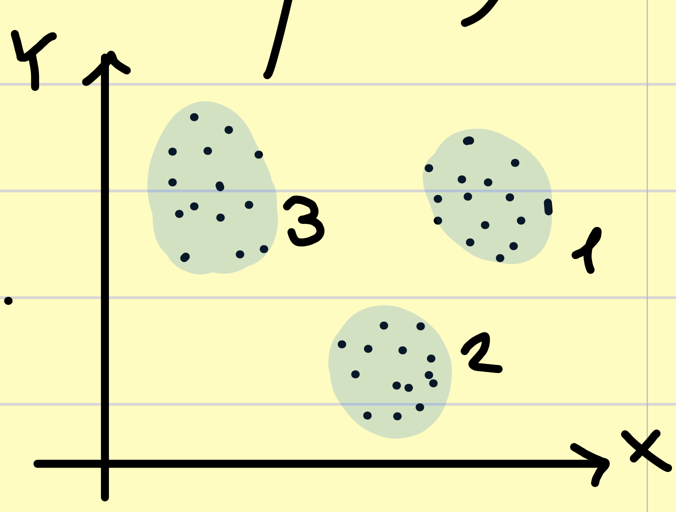
$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 \\ 0 & 2 & 0 & -1 & 0 \\ 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

K-means Clustering (Machine Learning)

This clustering algorythm builds groups out of data.

The hypothesis is that data are Euclidean (we can measure a distance btw. datapoints).

„K" is a parameter giving the number of groups.



(−). We need to tell the algorythm, how many groups we want.

(+). This algorythm is fast & efficient.

K-Means Algorythm:

Step 0. Decide the number of clusters "K".

Step 1. Distribute the datapoints in "K" groups.

Step 2. Find the "centroid" of each group.

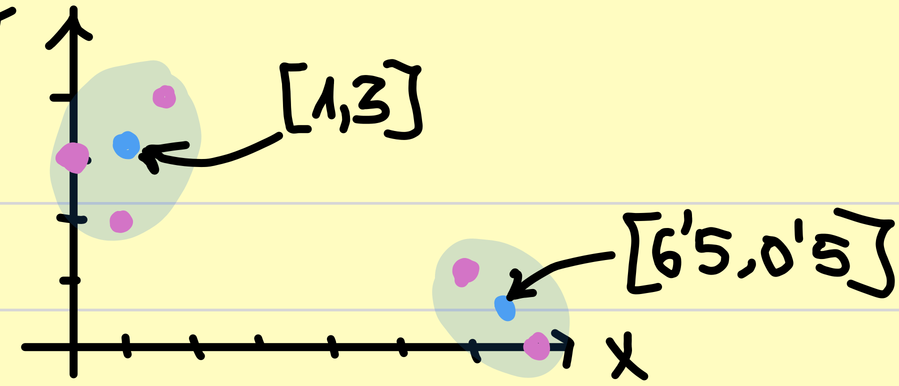
Step 3. Calculate the distance of all the points to all centroids.

Step 4. Cluster the points by their distance to the centroids.

Step 5. Repeat from Step 1 until the centroids are constant.

Excercise: we are given the location of 5 factories. Find out the best position of TWO warehouses that ought to deliver these factories.

$$\begin{array}{cc} & \begin{array}{ccccc} A & B & C & D & E \end{array} \\ X: & [0,6,1,7,2] \\ Y: & [3,1,2,0,4] \end{array}$$



· Step 0. $K=2$

· Step 1. $G1\ \{A,B,C\}$ $G2\ \{D,E\}$

· Step 2. Centroids: $z1=\left[\frac{0+6+1}{3},\frac{3+1+2}{3}\right]=[2'33,2]$

$z2=\left[\frac{7+2}{2},\frac{0+4}{2}\right]=[4'5,2]$

· Step 3. Distances:

$d_{A,Z1}=\sqrt{(0-2'33)^2+(3-2)^2}=2'53$; $d_{A,Z2}=\sqrt{(0-4'5)^2+(3-2)^2}=4'61$

$d_{B,Z1}=\sqrt{(6-2'33)^2+(1-2)^2}=3'8$; $d_{B,Z2}=\sqrt{(6-4'5)^2+(1-2)^2}=1'8$

$d_{C,Z1}=\sqrt{(1-2'33)^2+(2-2)^2}=1'33$; $d_{C,Z2}=\sqrt{(1-4'5)^2+(2-2)^2}=3'5$

$d_{D,Z1}=\sqrt{(7-2'33)^2+(0-2)^2}=5'08$; $d_{D,Z2}=\sqrt{(7-4'5)^2+(0-2)^2}=3'2$

$d_{E,Z1}=\sqrt{(2-2'33)^2+(4-2)^2}=2'02$; $d_{E,Z2}=\sqrt{(2-4'5)^2+(4-2)^2}=3'2$

Our new groups are: $G_1^*=[A,C,E]$; $G_2^*=[B,D]$

New centroids: $z_1^* = \left[\frac{0+1+2}{3}, \frac{3+2+4}{3}\right] = [1,3]$

$z_2^* = \left[\frac{6+7}{2}, \frac{1+0}{2}\right] = [6'5, 0'5]$

New distances:

$d_{A,z_1^*} = \sqrt{(0-1)^2+(3-3)^2} = 1 \quad < \quad d_{A,z_2^*} = \sqrt{(0-6'5)^2+(3-0'5)^2}$

$d_{C,z_1^*} = \sqrt{(1-1)^2+(2-3)^2} = 1 \quad < \quad d_{C,z_2^*} = \sqrt{(1-6'5)^2+(2-0'5)^2}$

$d_{E,z_1^*} = \sqrt{(2-1)^2+(4-3)^2} = \sqrt{2} \quad < \quad d_{E,z_2^*} = \sqrt{(2-6'5)^2+(4-0'5)^2}$

$d_{B,z_1^*} = \sqrt{(6-1)^2+(1-3)^2} = \sqrt{29} \quad > \quad d_{B,z_2^*} = \sqrt{(6-6'5)^2+(1-0'5)^2}$

$d_{D,z_1^*} = \sqrt{(7-1)^2+(0-3)^2} = 6 \quad > \quad d_{D,z_2^*} = \sqrt{(7-6'5)^2+(0-0'5)^2}$

Groups remain constant and we can close the Algorithm.

$G_1\{A,C,E\} \qquad G_2\{B,D\}$

$W_1\{1,3\} \qquad W_2\{6'5, 0'5\}$

Excercise: The position of 6 factories with different demands(*) on raw material

(*) weighted mean

is given by their coordinates [X,Y].
Every factory is delivered by one of two warehouses. To reduce the transport cost, the suppliers ask you to perform an analysis and recommend the optimal position of their supply points.

$$\begin{array}{cc} & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ X & [\,1,\ 2,\ 0,\ 6,\ 7,\ 3\,] \\ Y & [\,3,\ 2,\ 1,\ 1,\ 2,\ 3\,] \\ \text{Demand} & [\,2,\ 1,\ 3,\ 1,\ 3,\ 1\,] \end{array}$$

Start with groups: $G_1[A,B,C]$ $G_2[D,E,F]$

hint:

$$Z_1[A,B,C] = \left[\frac{1.2+2.1+0.3}{3}, \frac{3.2+2.1+1.3}{3}\right]$$