

Kunstig intelligens

---

Kunstig intelligens

---

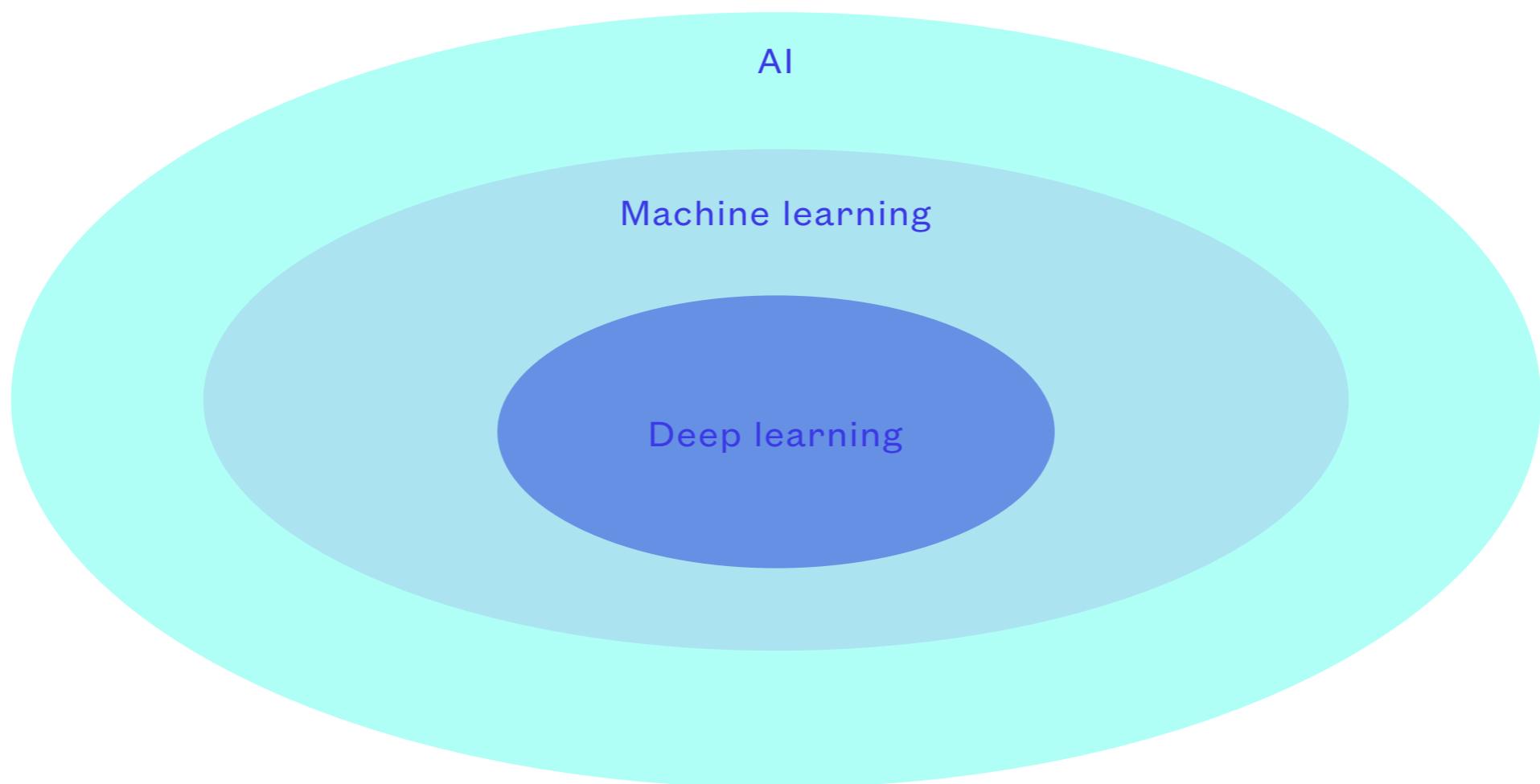
Choose wisely

---

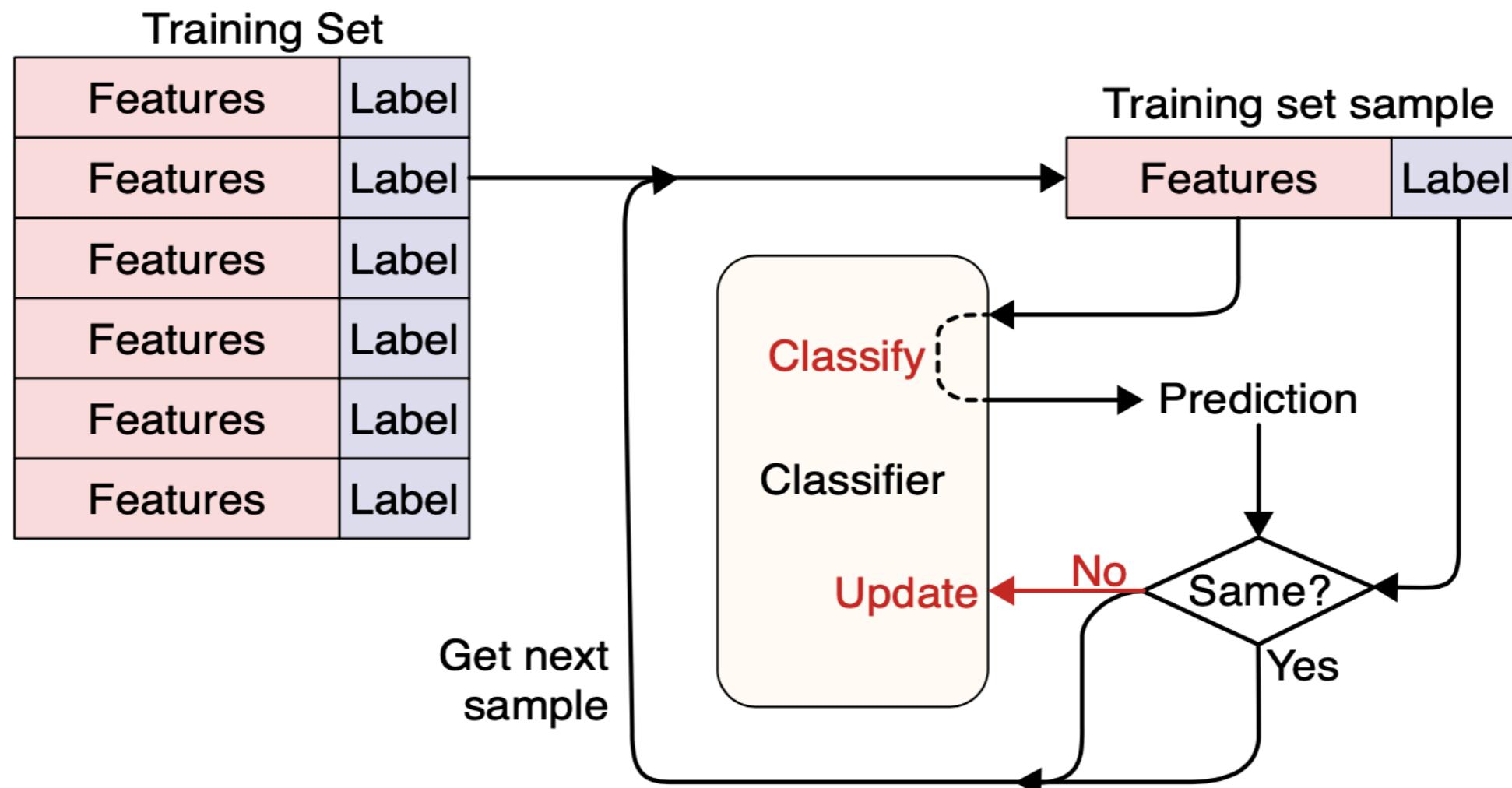
# Maskinlæring

---

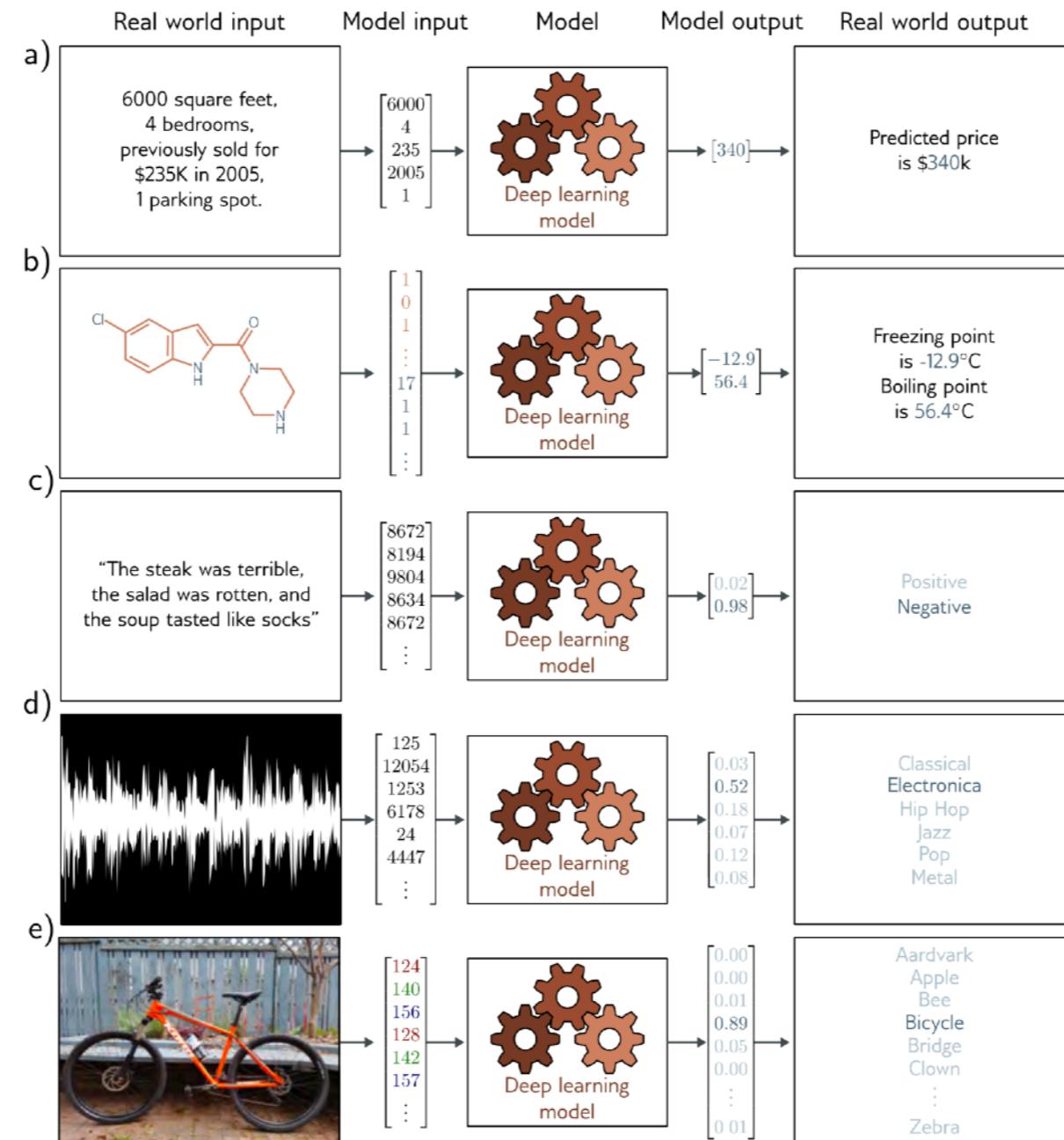
**REPETISJON**



## REPETISJON



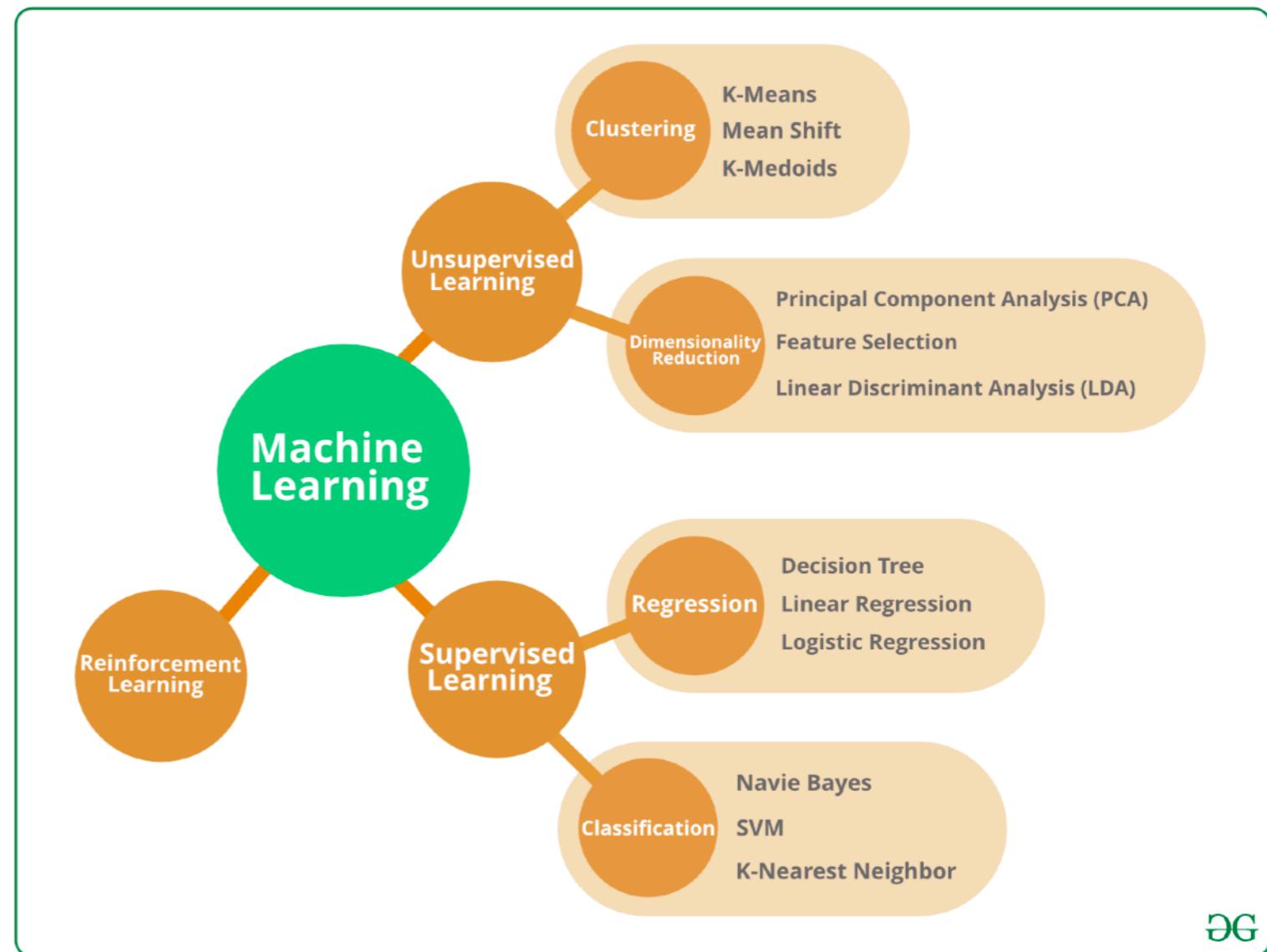
## REPETISJON



**Figure 1.2** Regression and classification problems. a) This *regression* model takes a vector of numbers that characterize a property and predicts its price. b) This *multivariate regression* model takes the structure of a chemical molecule and predicts its melting and boiling points. c) This *binary classification* model takes a restaurant review and classifies it as either positive or negative. d) This *multiclass classification* problem assigns a snippet of audio to one of  $N$  genres. e) A second multiclass classification problem in which the model classifies an image according to which of  $N$  possible objects it might contain.

# Kategorier

- Supervised
- Unsupervised
- Reinforcement



# Supervised

- Linear regression
- Logistic regression
- Decision trees
- Random forests
- Support Vector Machines
- Naive Bayes
- K-Nearest Neighbors



Bird



Forest

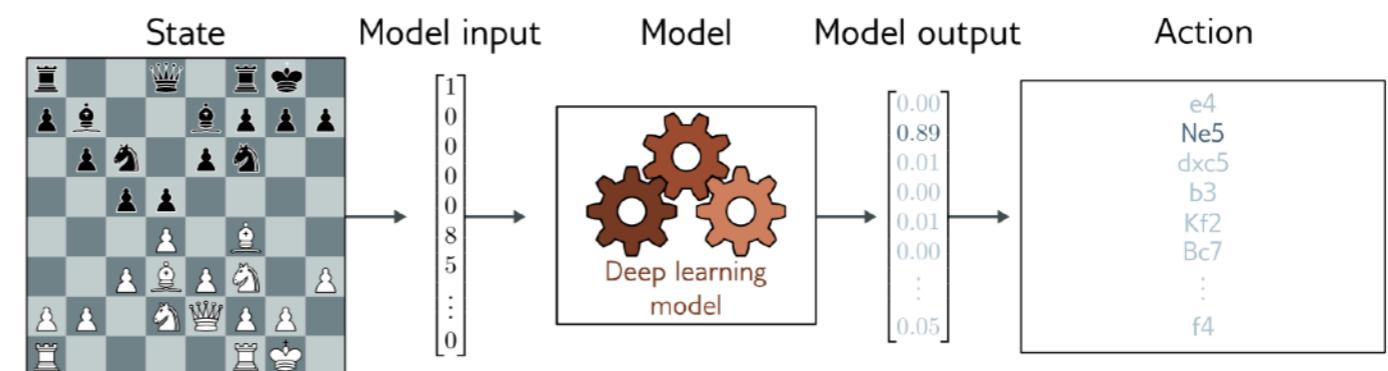
# Unsupervised

---

- K-Means clustering
  - Hierarchical clustering
  - DBSCAN (Density-based spatial clustering of applications with noise)
  - PCA
  - Association rules
  - Anomaly detection
-

# Reinforcement

- Q-learning
- Deep-Q networks
- Policy gradient methods
- Actor-critic methods
- Monte Carlo tree search

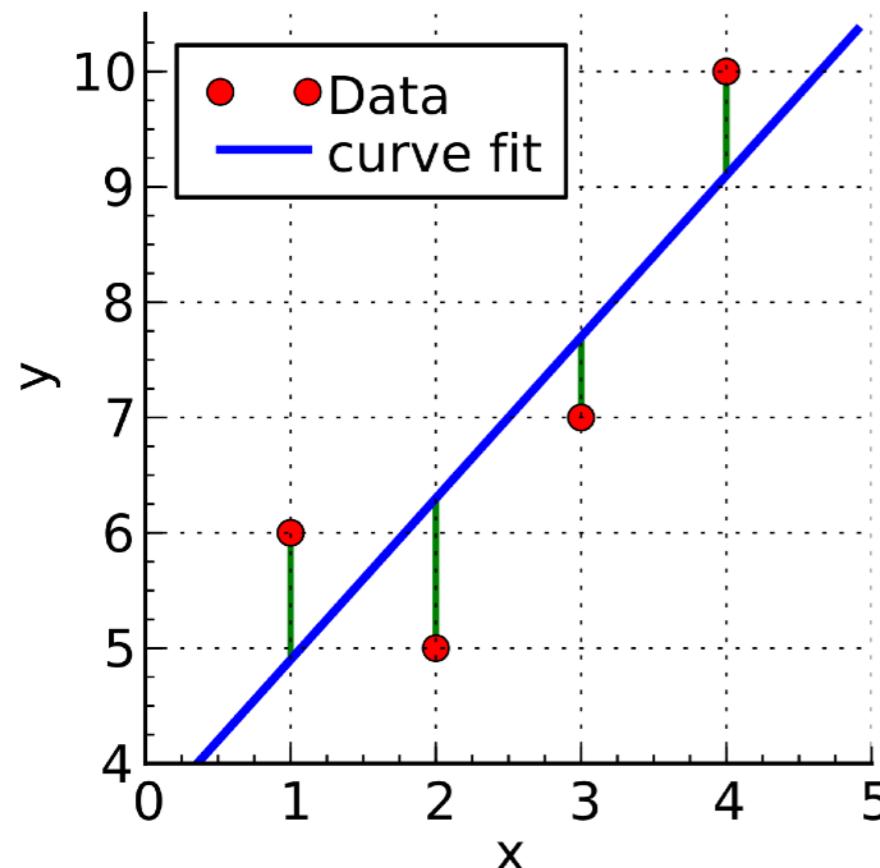


**Figure 1.13** Policy networks for reinforcement learning. One way to incorporate deep neural networks into reinforcement learning is to use them to define a mapping from the state (here position on chessboard) to the actions (possible moves). This mapping is known as a *policy*. Adapted from Pablok (2017).

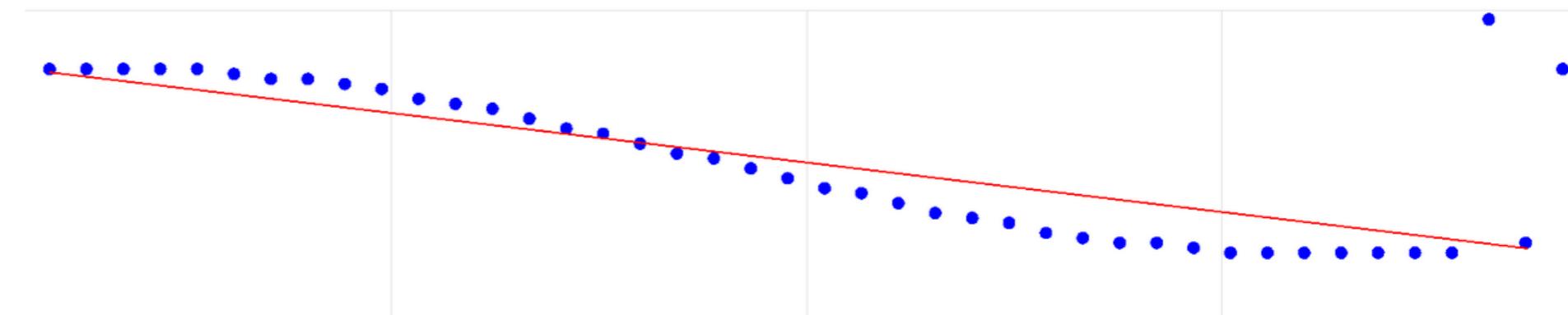
CC-BY-NC-ND license. (C) MIT Press.

# Linear regression

SUPERVISED



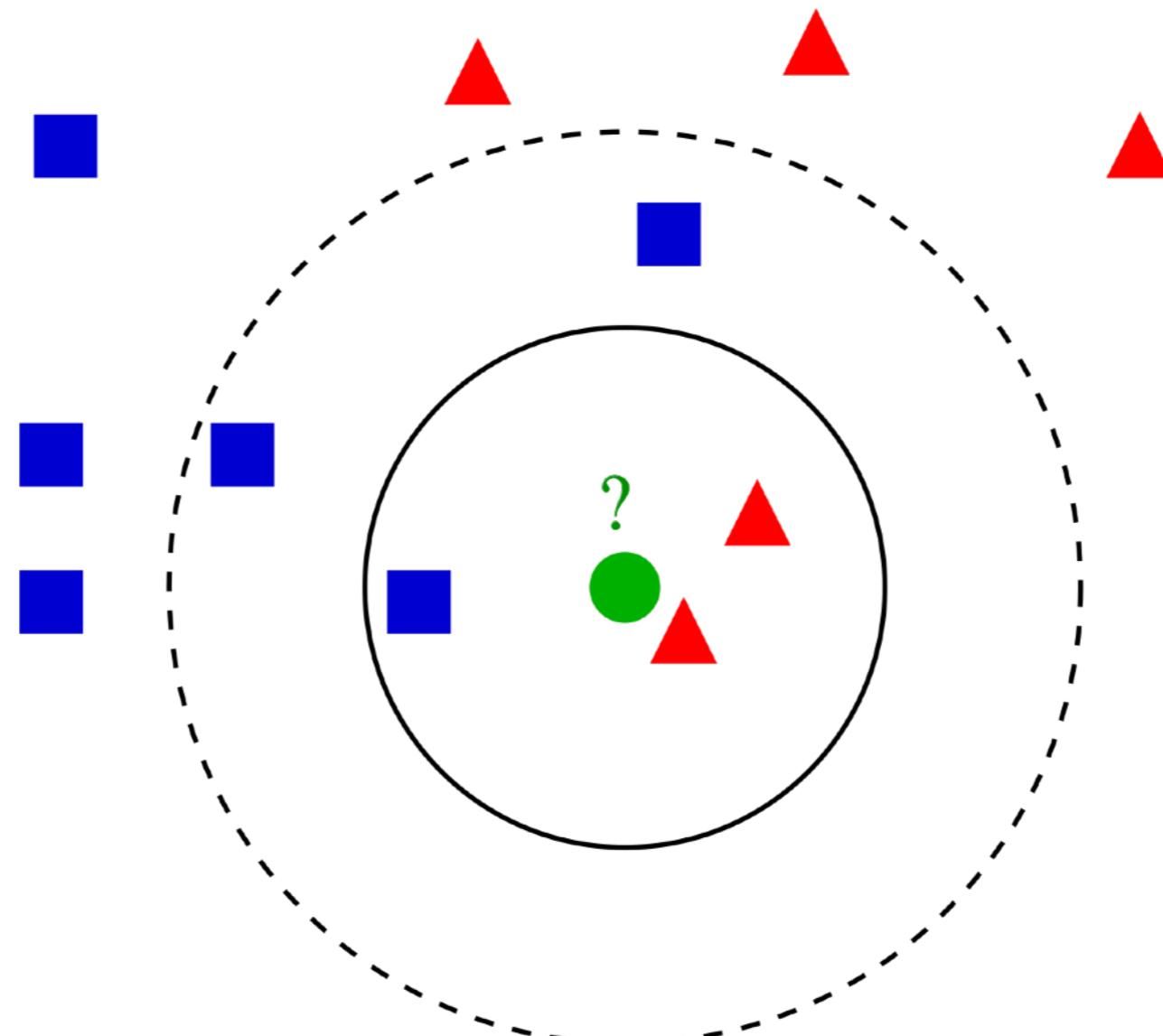
REPETISJON



# *k*-nearest-neighbors

SUPERVISED

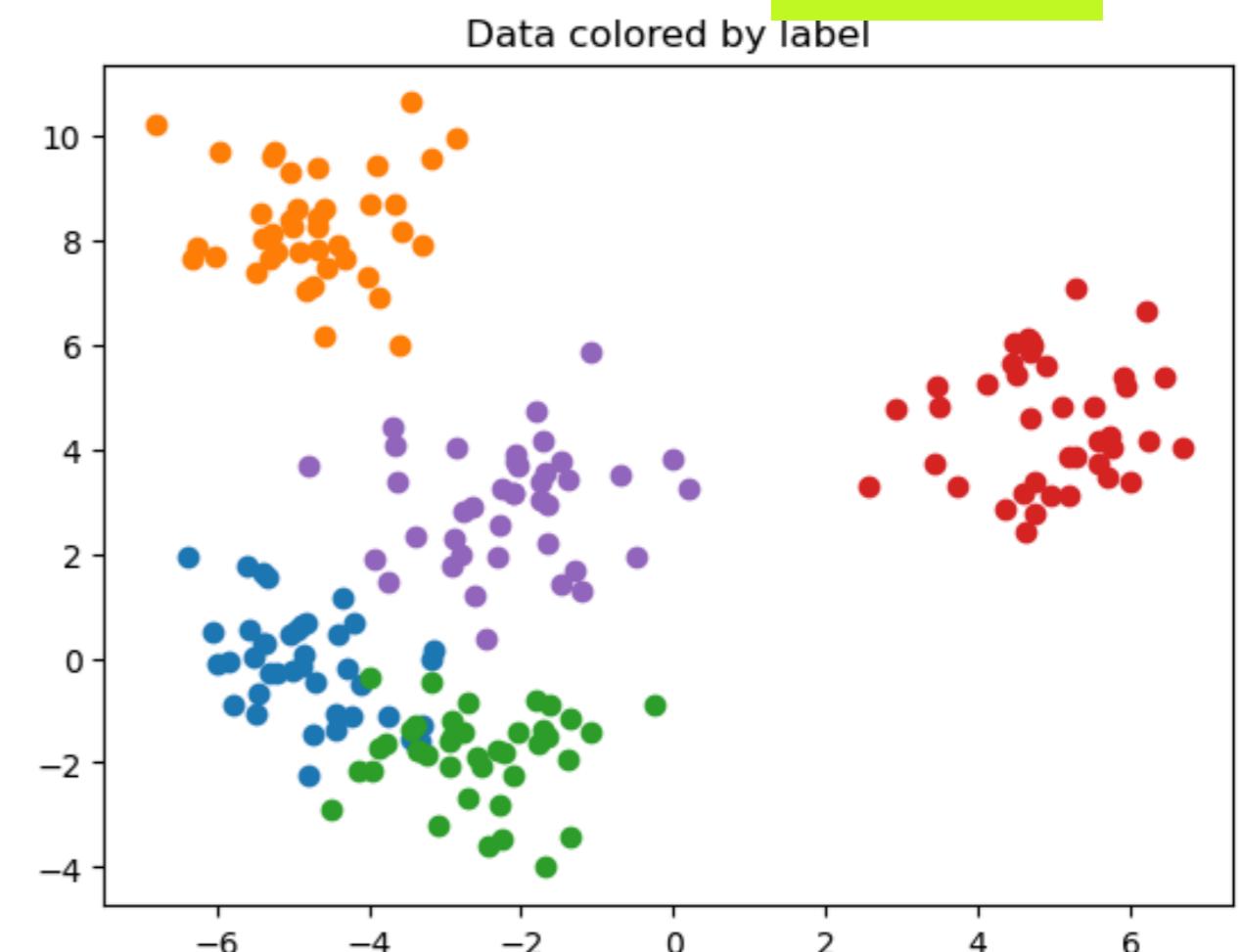
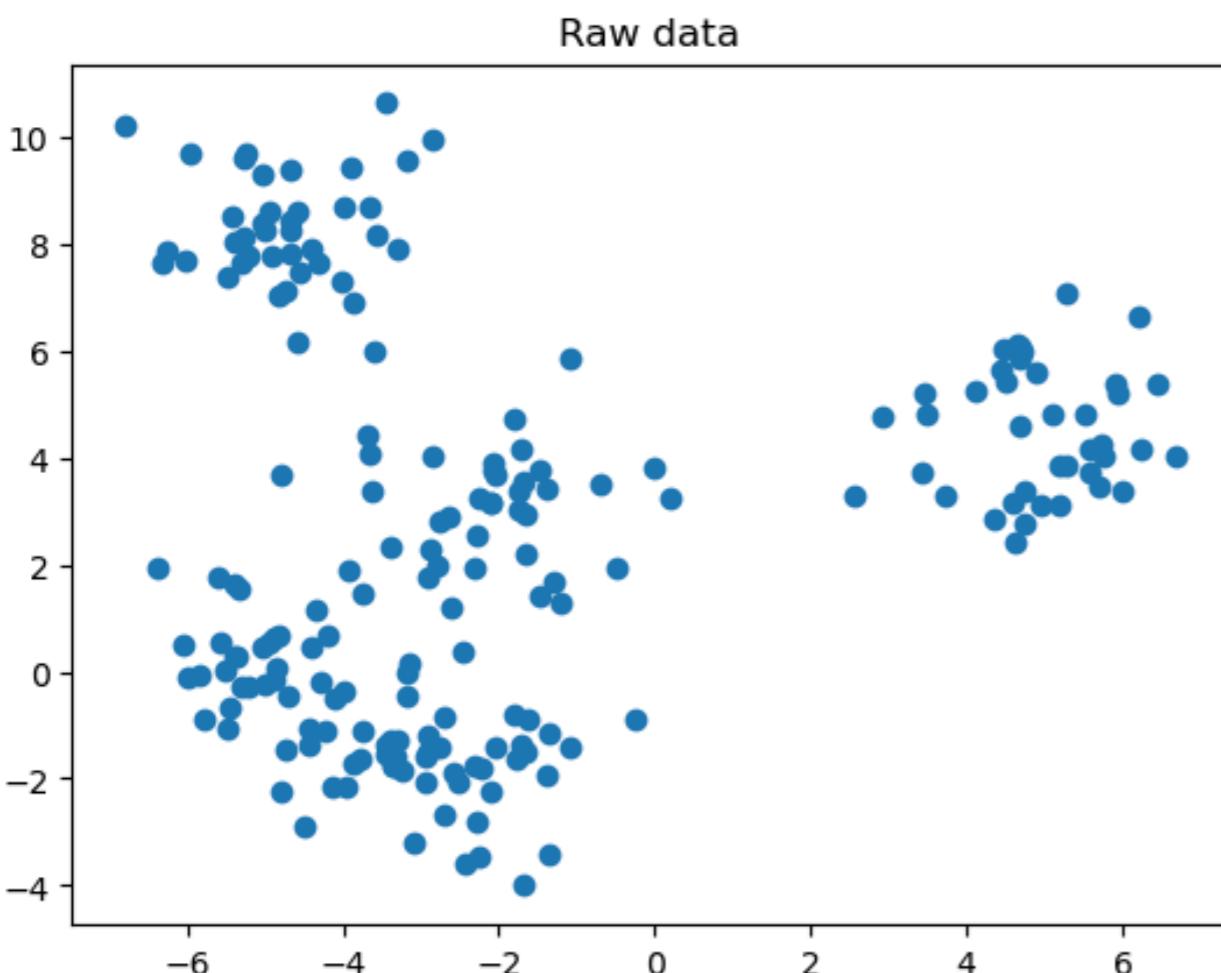
REPETISJON



# **k-means clustering**

**UNSUPERVISED**

**REPETISJON**



# **k-means clustering**

**UNSUPERVISED**

**REPETISJON**

- 1: Spesifisert antall cluster ( $k$ )
- 2: Initialiser  $k$  centroids
- 3: gjenta:
- 4: Forventing - Tildel kvart punkt til den næreste centroiden.
- 5: Maksimer - Beregn ny centroid (mean) for hvert cluster.
- 6: fremtil: centroidene sin plassering ikke endrer seg.

---

# Faktorer

- Nøyaktighet
  - Kompleksitet
  - Skalerbarhet
  - Robusthet
  - Fleksibilitet
  - Effektivitet
-

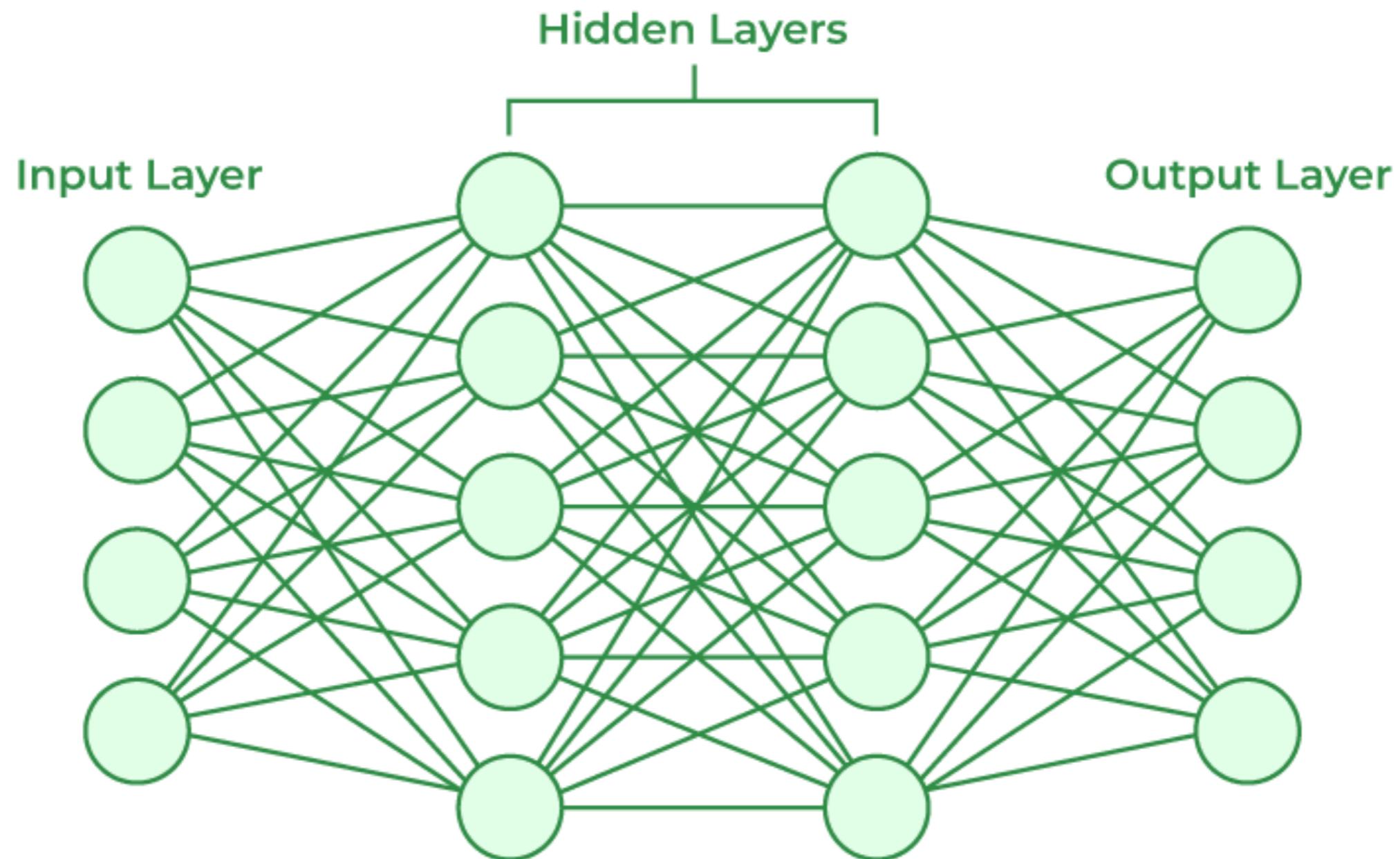
De svarte boksene

---

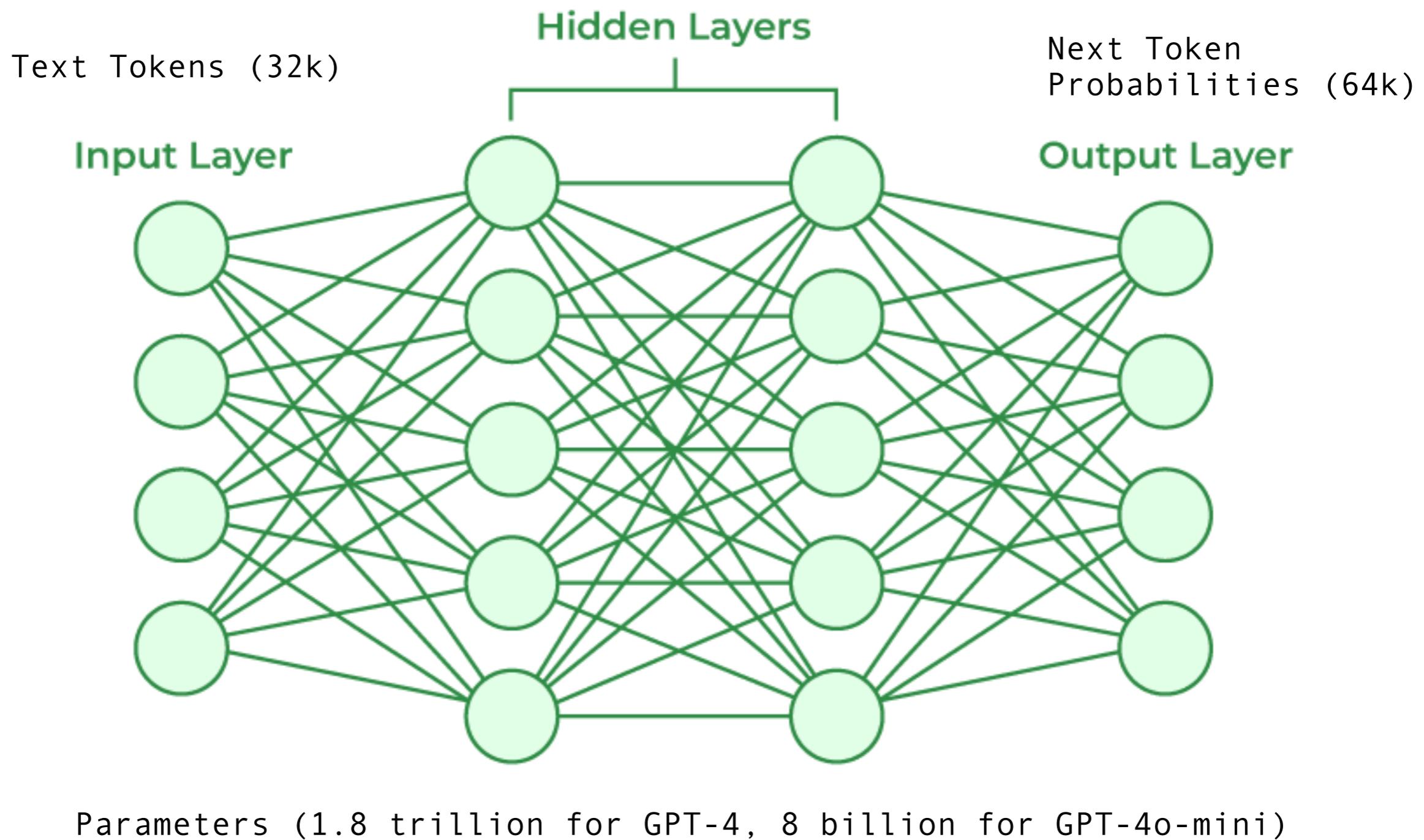
Nevrale nettverk

---

# Neural network



# LLM



# Tokenizer

GPT-4o & GPT-4o mini    GPT-3.5 & GPT-4    GPT-3 (Legacy)

Dette er en tokenizer test.

**Clear** **Show example**

Tokens	Characters
6	27

Dette **er** en **tokenizer** test.

The screenshot shows the OpenAI Tokenizer interface. At the top, there are three tabs: 'GPT-4o & GPT-4o mini' (selected), 'GPT-3.5 & GPT-4', and 'GPT-3 (Legacy)'. Below the tabs is a text input field containing the sentence 'Dette er en tokenizer test.'. Underneath the input field are two buttons: 'Clear' and 'Show example'. Below these buttons is a table with two columns: 'Tokens' and 'Characters'. The 'Tokens' column contains the number '6', and the 'Characters' column contains the number '27'. At the bottom of the interface is a visual representation of the tokens, where each word is highlighted with a different color: 'Dette' (purple), 'er' (green), 'en' (orange), 'tokenizer' (red), and 'test.' (blue).

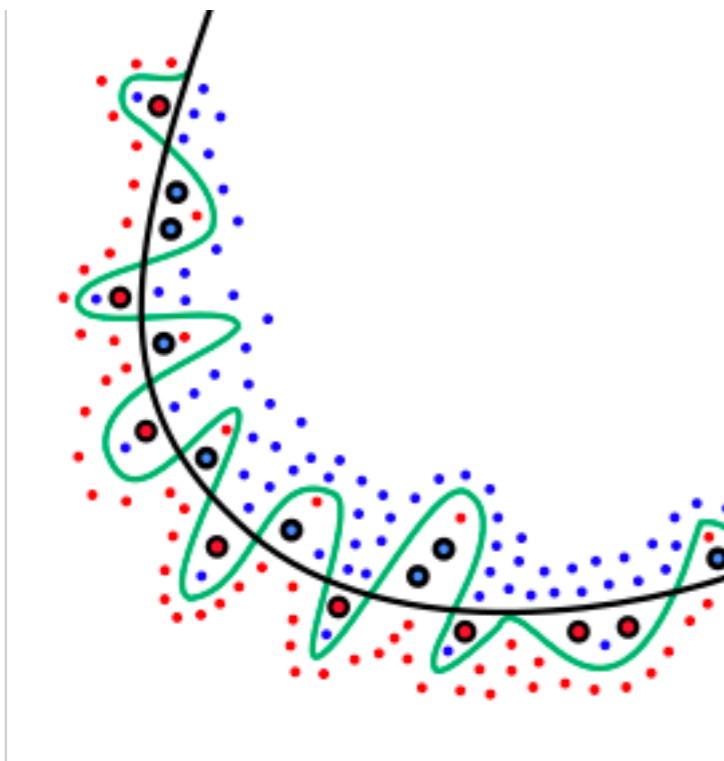
<https://platform.openai.com/tokenizer>

---

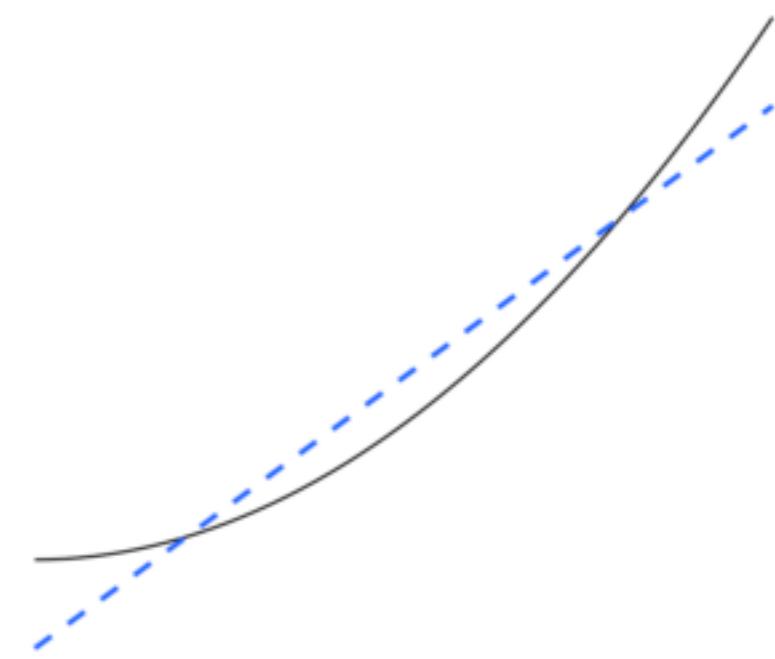
# Ymse

---

# Overfitting / Underfitting



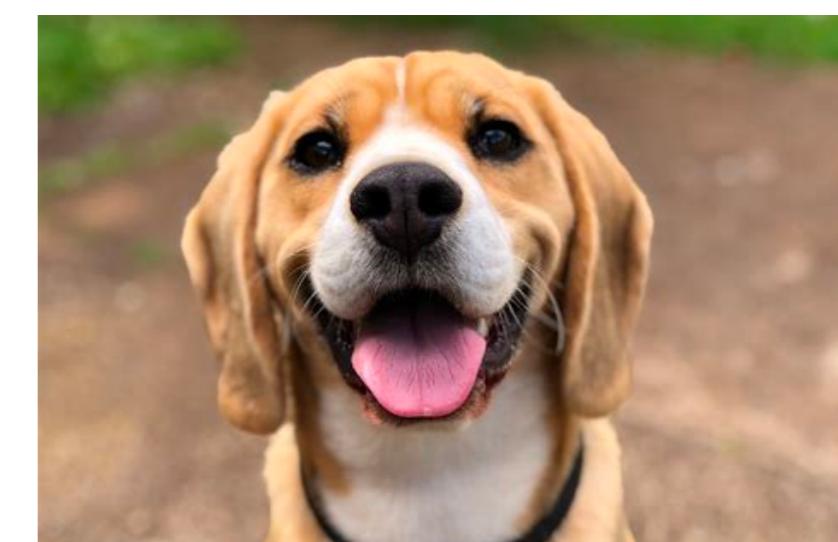
Overfitted



Underfitted

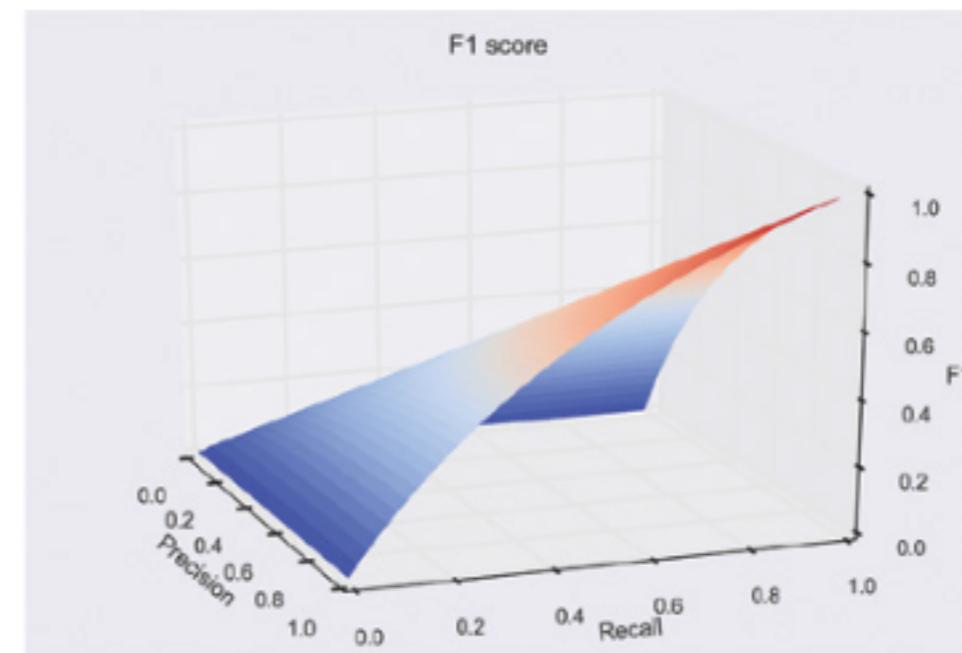
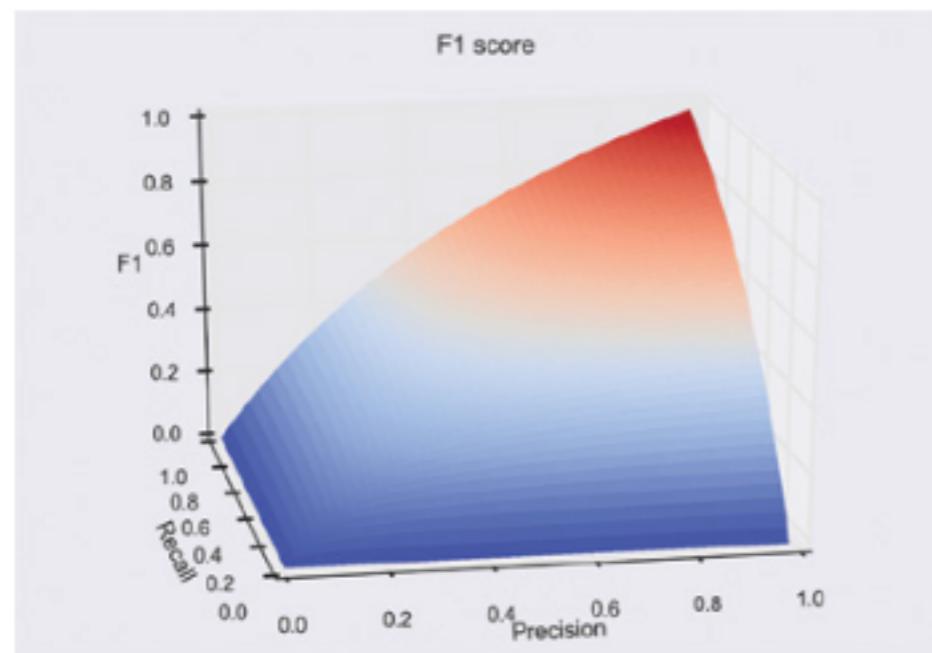
# Overfitting

---



# Performance

		Prediction
		Positive
		Negative
Actual value	Positive	True positive ○ ○ ○ ○ ○ ○
	Negative	False negative ○ ○ ○ ○
Actual value	Positive	False positive ■ ■
	Negative	True negative ■ ■ ■ ■ ■ ■ ■ ■



# Performance

Accuracy:

Total-Positive + Total-Negative / Total-Observation

$$(6 + 8) / 20 = 0,7$$

Precision:

Correct-Marked-Positive / Marked-Positive

$$6 / (6+2) = 0,75$$

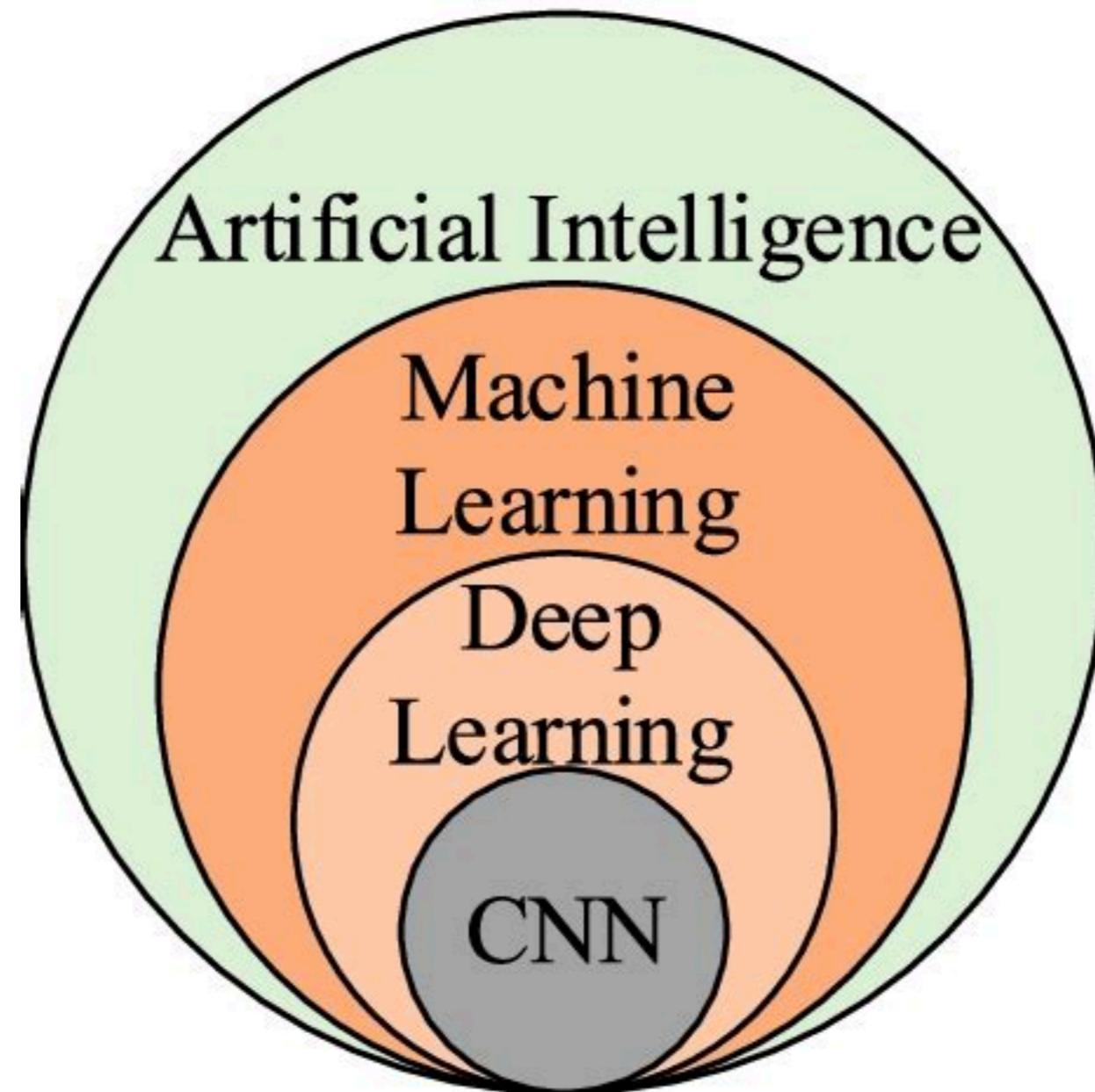
Recall / Hit-rate:

Marked-Positive / Total-Positive

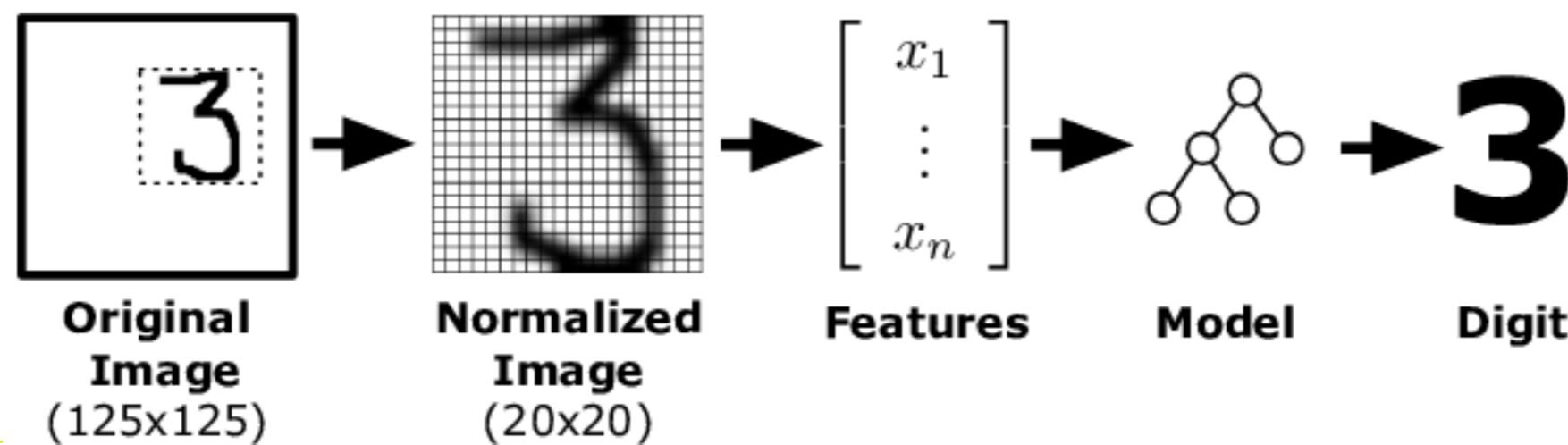
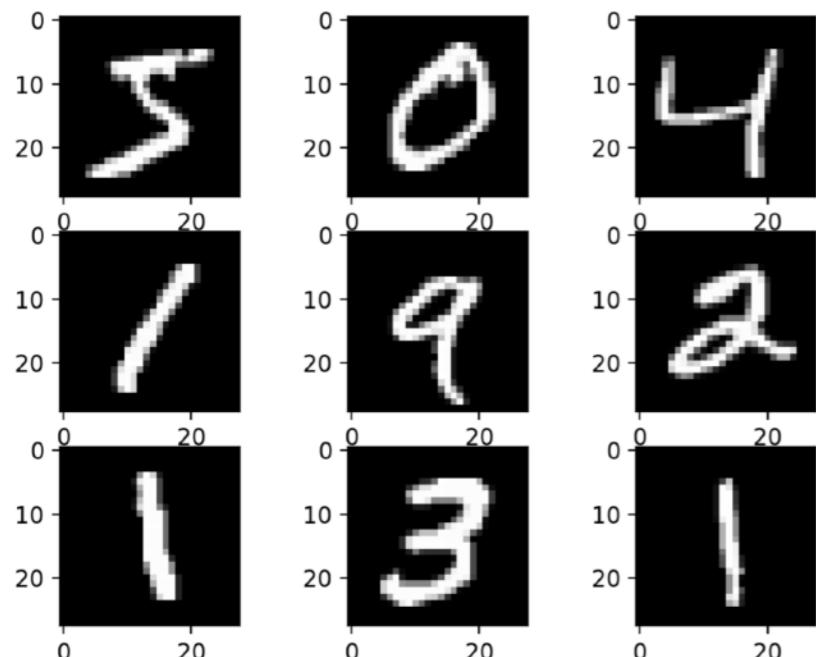
$$6 / (6+4) = 0,6$$

		Prediction
		Positive
		Negative
Actual value	Positive	True positive ○ ○ ○ ○ ○ ○
	Negative	False negative ○ ○ ○ ○ False positive ■ ■
		True negative ■ ■ ■ ■ ■ ■ ■ ■

# CNN



# Handwritten digits

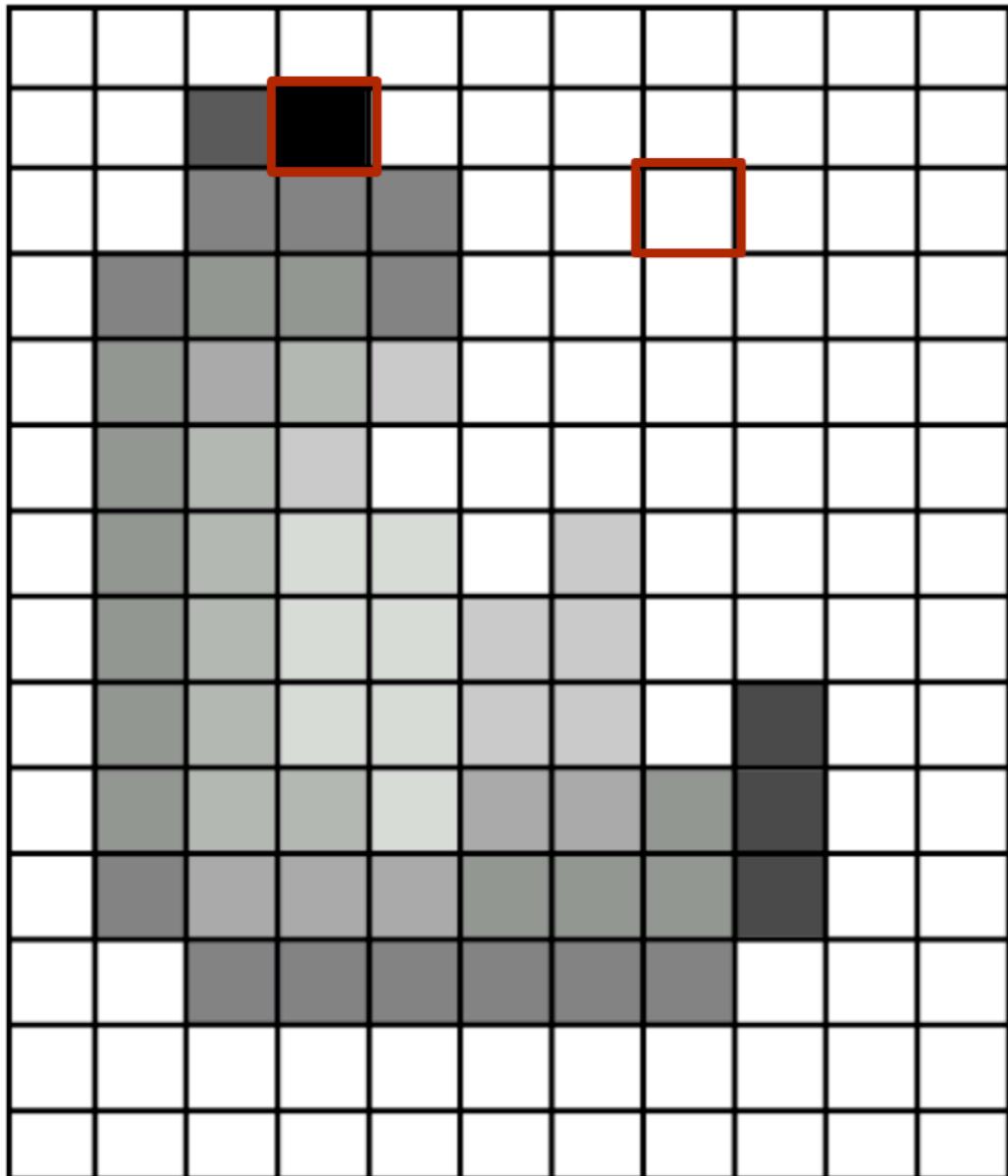


La oss se hva vi får til

---

# Bildebehandling

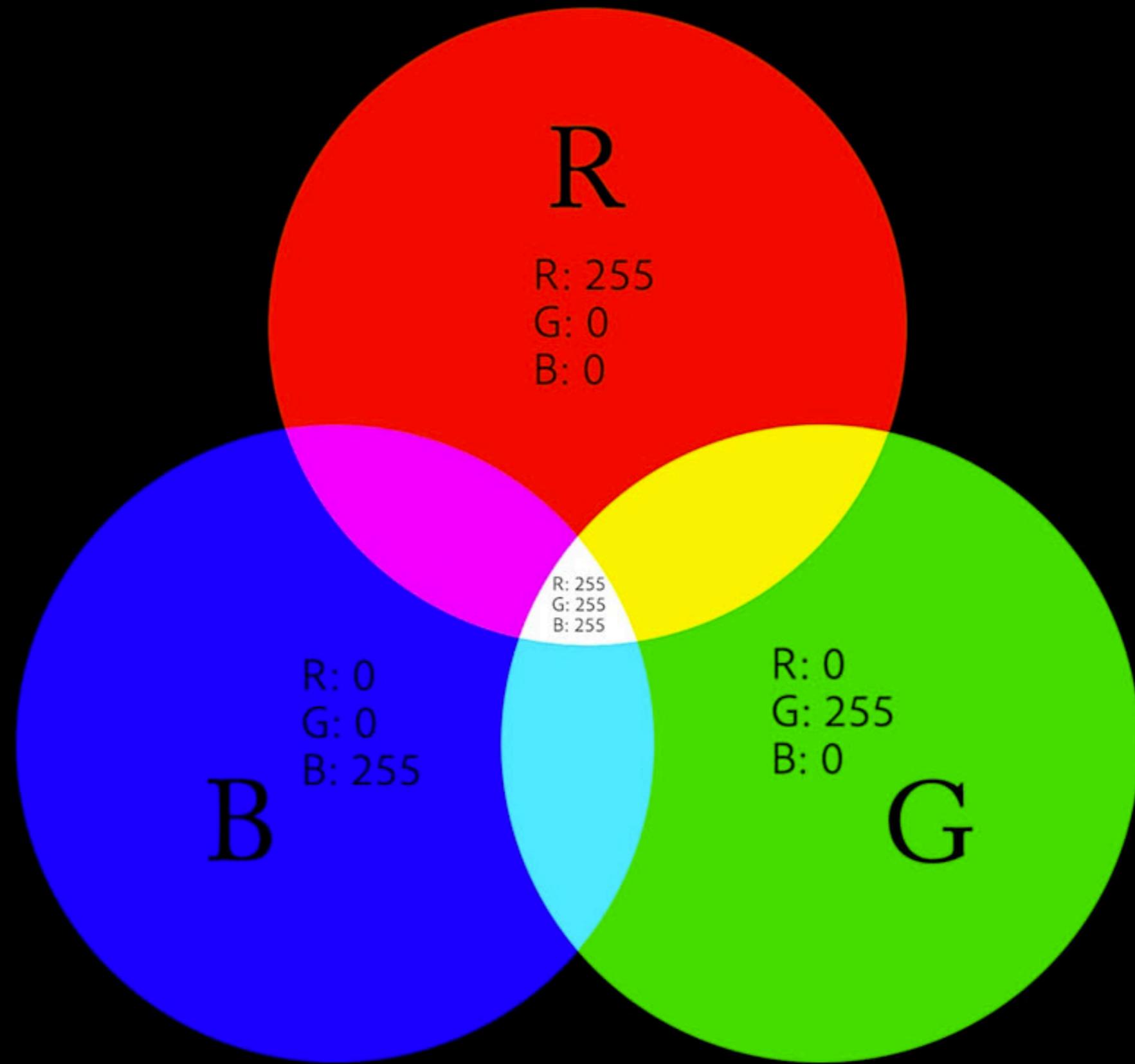
---

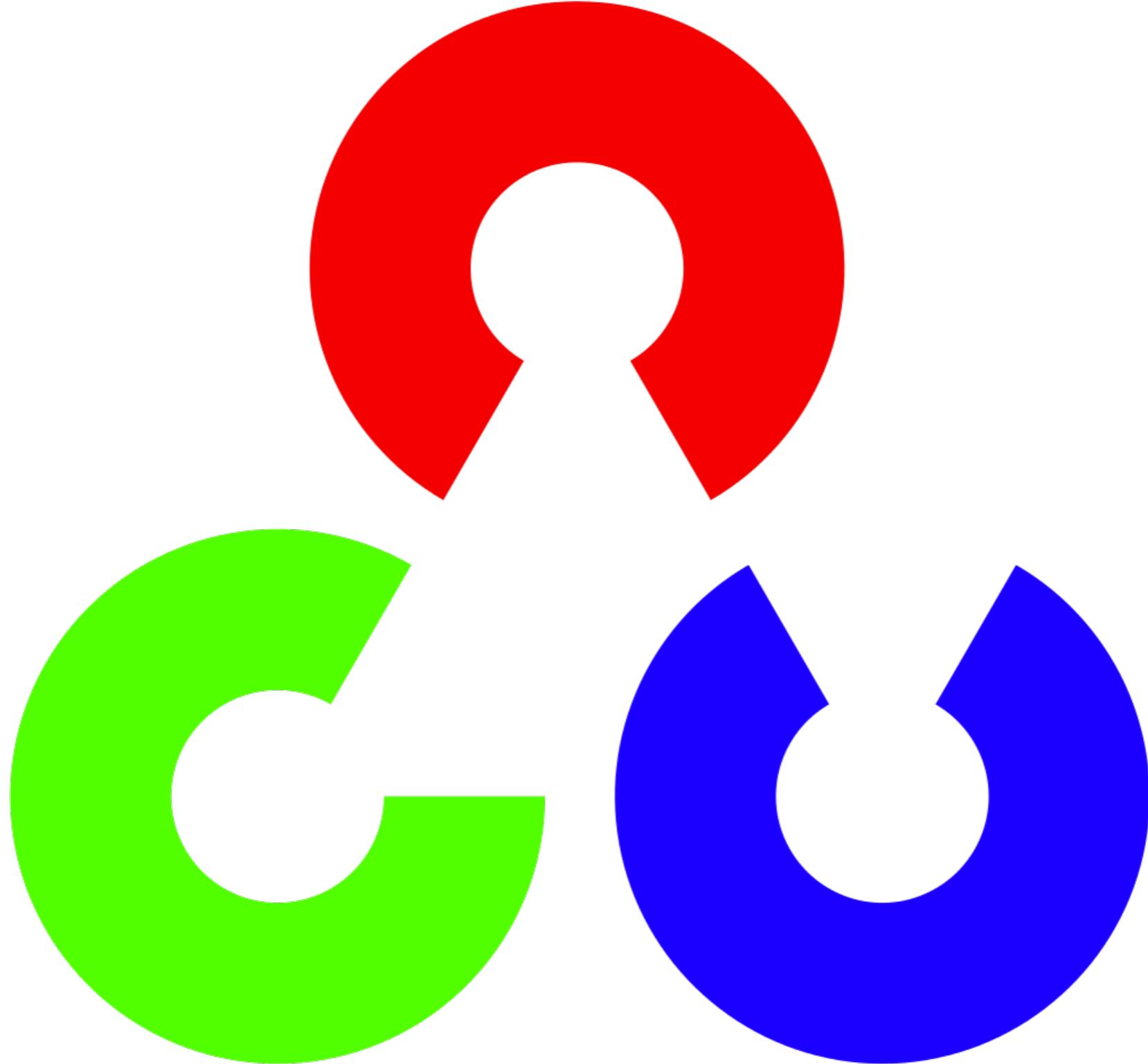


11 piksler											
255	255	255	255	255	255	255	255	255	255	255	255
255	255	20	0	255	255	255	255	255	255	255	255
255	255	75	75	255	255	255	255	255	255	255	255
255	75	95	95	75	255	255	255	255	255	255	255
255	96	127	145	175	255	255	255	255	255	255	255
255	127	145	175	175	175	255	255	255	255	255	255
255	127	145	200	200	175	175	95	255	255	255	255
255	127	145	200	200	175	175	95	47	255	255	255
255	127	145	145	175	127	127	95	47	255	255	255
255	74	127	127	127	95	95	95	47	255	255	255
255	255	74	74	74	74	74	74	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

**0 = black; 255 = white**

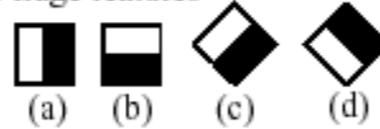
R: 0  
G: 0  
B: 0



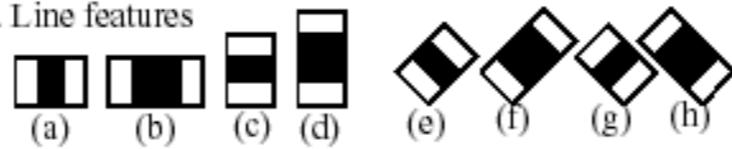


**OpenCV**

1. Edge features



2. Line features



3. Center-surround features

