



# Übungslektion 14 – Machine Learning III

Informatik II

27. / 28. Mai 2025

# Heutiges Programm

- Convolutional Neural Networks CNNs
- K-means Clustering
- Dimensionsreduktion
- Kahoot!

# 1. Recap: Convolutional Neural Networks (CNNs)

---

# Warum CNNs für Bilder?

- Bilder besitzen spezielle Eigenschaften: **Lokalität**, **Invarianz** und **Hierarchie**.
- Lokale Bildbereiche enthalten relevante Merkmale (z.B. Kanten, Muster).
- Objekte im Bild bleiben auch bei Verschiebung erkennbar (Translation-Invarianz).
- Komplexe Strukturen entstehen aus einfachen lokalen Mustern.

# Wie werden Bilder gespeichert?

- Bilder bestehen aus Pixelwerten, oft mit mehreren Farbkanälen (RGB).
- Ein Bild ist als mehrdimensionales Array im Computer gespeichert.
- Beispiel: Graustufenbild (1 Kanal), Farbbild (3 Kanäle).
- Bilddaten können groß sein – effiziente Verarbeitung ist wichtig!

# Faltung (Convolution): Grundidee

- **Faltungsschichten** wenden kleine Filter (Kerne) lokal auf das Bild an.
- Jeder Filter erkennt ein bestimmtes Muster, z.B. eine Kante oder Textur.
- Das Resultat ist eine *Feature Map*, die zeigt, wo das Muster gefunden wurde.
- Mehrere Filter führen zu mehreren Feature Maps (Kanäle).

The diagram illustrates the convolution operation. On the left is the **Input Bild**, an 8x8 grid of values. A green diagonal highlights the values 9, 1, 9, 1, 9, 1, 9, 9. Below it is the label "Input Bild" and "Padding (oft mit 0)". In the middle is the **Kernel**, a 3x3 grid with values [-1, 0, 1] repeated in each row. Below it is the label "Kernel". To the right of the kernel is a star symbol  $\star$  and an equals sign  $=$ . On the far right is the resulting **Feature Map**, an 8x8 grid of values ranging from -24 to 16, with a color gradient from yellow to dark purple. Below it is the label " $I * K$ " and "feature map".

0	0	0	0	0	0	0	0
0	1	1	9	1	1	0	0
0	1	9	9	1	1	0	0
0	9	1	9	1	1	0	0
0	1	1	9	1	1	0	0
0	1	1	9	1	1	0	0
0	1	1	9	1	1	0	0
0	9	9	9	9	9	0	0
0	0	0	0	0	0	0	0

Input Bild  
Padding (oft mit 0)

-1	0	1
-1	0	1
-1	0	1

Kernel

$\star$        $=$

10	16	-8	-16	-2			
11	16	-8	-24	-3			
11	16	-8	-24	-3			
3	16	0	-24	-3			
3	24	0	-24	-3			
11	16	0	-16	-11			
10	8	0	-8	-10			

$I * K$   
"feature map"

# Mehrkanalige Faltung und Padding

- Bei Bildern mit mehreren Kanälen wird für jeden Kanal ein separater Filter genutzt.
- Die Anzahl der Parameter bleibt niedrig, unabhängig von der Bildgröße.
- **Padding** wird oft verwendet, damit die Feature Map dieselbe Größe wie das Eingabebild behält (meist mit Nullen).

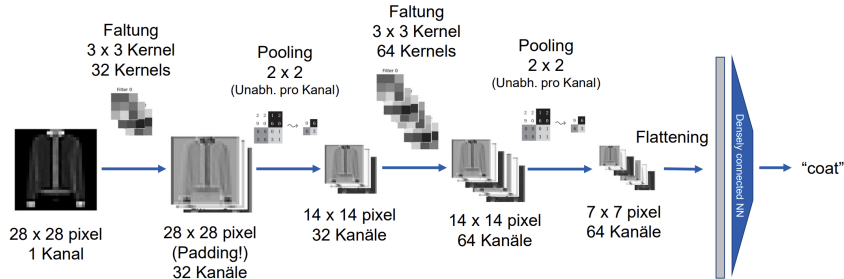
# Pooling (Downsampling)

- Pooling reduziert die räumliche Größe der Feature Maps (z.B. Max-Pooling).
- Sorgt für kompaktere Darstellung, weniger Rechenaufwand und bessere Generalisierung.
- Max-Pooling wählt den größten Wert aus einem kleinen Fenster (z.B. 2x2).



# Architektur eines CNNs: Beispiel

- Ein typisches CNN besteht aus mehreren Faltungs- und Pooling-Schichten, gefolgt von voll verbundenen Schichten.
- Beispiel (FashionMNIST):
  - 2 Faltungsschichten (z.B. 32 und 64 Filter)
  - 2 Max-Pooling-Schichten
  - Flattening Dense-Layer (128 Neuronen) Ausgabe-Layer (z.B. Softmax)



# Zusammenfassung: CNNs

- CNNs sind für Bilder besonders geeignet, weil sie Lokalität, Hierarchie und Invarianz ausnutzen.
- Faltung extrahiert lokale Merkmale; Pooling verdichtet Informationen.
- Parametereffizient durch lokale Filter.
- Geeignet für Aufgaben wie Bildklassifikation, Objekterkennung und mehr.

## 2. Recap: K-Means Clustering

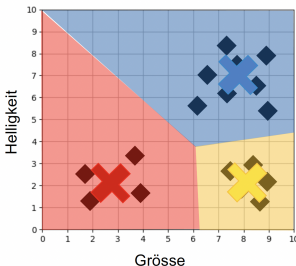
---

# Was ist K-Means?

- Ziel: Teile  $n$  Datenpunkte  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  in  $K$  Cluster ein.
- Jedem Cluster  $k$  ist ein Zentrum  $\boldsymbol{\mu}_k$  zugeordnet.
- Jeder Punkt wird dem nächstgelegenen Zentrum zugewiesen:

$$c_i = \arg \min_{k=1, \dots, K} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

- $K$  ist ein Hyperparameter.

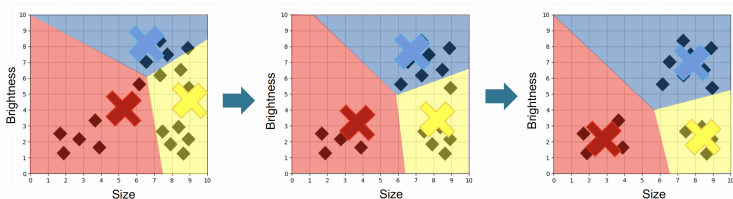


# K-Means Training

1. Initialisiere Clusterzentren  $\mu_1, \dots, \mu_K$  zufällig.
2. **Zuweisungsschritt:** Weise jedem Punkt das nächste Zentrum zu.
3. **Update-Schritt:** Berechne für jedes Cluster das neue Zentrum als Mittelwert:

$$\mu_k = \frac{1}{|C_k|} \sum_{i:c_i=k} \mathbf{x}_i$$

4. Wiederhole Schritt 2 und 3 bis zur Konvergenz.



# Eigenschaften und Herausforderungen von K-Means

- K-Means bestimmt nicht automatisch die beste Anzahl  $K$ .
- Sensitiv gegenüber der Wahl der Anfangszentren.
- Kann zu Überanpassung führen, wenn  $K$  zu groß gewählt wird.
- Clusterzentren müssen keine echten Datenpunkte sein.
- **Konvergiert immer** (aber nicht immer zu einem globalen Optimum)

# K-Means Visualisierung

K-Means Clustering

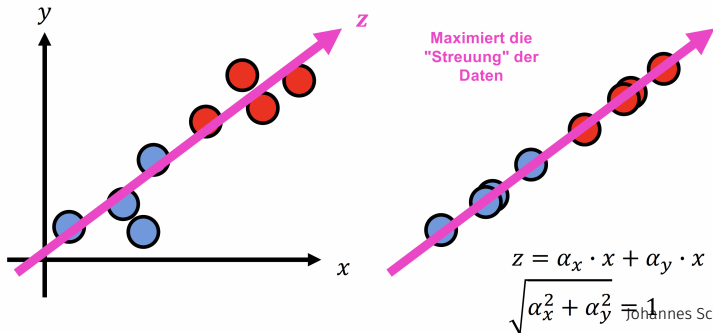
### 3. Recap: Dimensionsreduktion

---



# Was ist Dimensionsreduktion?

- Ziel:  $D$ -dimensionale Daten  $\mathbf{x}_i \in \mathbb{R}^D$  auf  $d < D$  Dimensionen projizieren.
- Formell: Finde Projektion  $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$ ,  $\mathbf{W} \in \mathbb{R}^{D \times d}$ .
- **Vorteile:** Kompakt, schnelleres Training, bessere Visualisierung, weniger Überanpassung.



# Lineare Dimensionsreduktion

## Lineare Methoden:

- **Prinzip:** Finden eine lineare Projektion des Datensatzes in einen Raum niedrigerer Dimension.
- **Beispiele:** Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA).
- **Eigenschaften:**
  - Nur lineare Beziehungen zwischen Variablen werden abgebildet.
  - Schnell und effizient, aber limitiert, wenn die Daten auf gekrümmten Mannigfaltigkeiten liegen.

## Nichtlineare Methoden:

- **Prinzip:** Abbildung der Daten auf eine niedrigdimensionale Mannigfaltigkeit, die auch nichtlineare Beziehungen erfasst.
- **Beispiele:** t-SNE, UMAP, Isomap, Locally Linear Embedding (LLE).
- **Eigenschaften:**
  - Erkennen komplexe, gekrümmte Strukturen im Datensatz.
  - Oft rechenintensiver und schwieriger zu interpretieren.

# Wann welche Methode verwenden?

## **Lineare Methoden:**

- Gut geeignet, wenn die wichtigsten Strukturen in den Daten linear sind.
- Schnelle Vorverarbeitung für Visualisierung, Kompression oder maschinelles Lernen.

## **Nichtlineare Methoden:**

- Besser, wenn komplexe Strukturen oder Cluster erwartet werden, die durch lineare Methoden nicht erfasst werden.
- Häufig in der explorativen Datenanalyse und Visualisierung (z.B. t-SNE für High-Dimensional Data).

## **Praxis:**

- Oft werden zuerst lineare Methoden wie PCA ausprobiert; bei Bedarf folgen nichtlineare Methoden.
- Interpretierbarkeit und Rechenaufwand sollten mitbedacht werden.

# Principal Component Analysis (PCA): Mathematik

- Finde Richtung  $\mathbf{u}_1$  mit maximaler Varianz:

$$\mathbf{u}_1 = \arg \max_{\|\mathbf{u}\|=1} \text{Var}(\mathbf{u}^T \mathbf{x})$$

- Maximiert wird  $\mathbf{u}^T S \mathbf{u}$ , wobei  $S$  die Kovarianzmatrix ist:

$$S = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

- $\mathbf{u}_1$  ist der Eigenvektor von  $S$  mit dem größten Eigenwert.

# PCA: Projektion und Rekonstruktion

- Projektion eines Punktes  $\mathbf{x}_i$  auf die erste Hauptkomponente:

$$z_i = \mathbf{u}_1^T (\mathbf{x}_i - \bar{\mathbf{x}})$$

- Für mehrere Komponenten:  $\mathbf{z}_i = \mathbf{U}^T (\mathbf{x}_i - \bar{\mathbf{x}})$ , wobei  $\mathbf{U}$  die Matrix der ersten  $d$  Eigenvektoren ist.
- Die projizierten Daten können für Visualisierung und weiteres Lernen genutzt werden.

# Eigenschaften und Grenzen von PCA

- PCA kann Dimensionen ohne signifikanten Informationsverlust reduzieren.
- Die wichtigsten Hauptkomponenten sind zueinander orthogonal.
- Funktioniert am besten, wenn die Merkmale linear korreliert sind.
- Nachteile: Nur lineare Zusammenhänge werden abgebildet, Skalierung der Daten kann wichtig sein.

## 4. Übung

---



# Übung: CNN-Kanäle und Padding

Betrachten Sie ein RGB-Bild mit Breite  $W = 320$ , Höhe  $H = 240$  und  $C = 3$  Kanälen (RGB). Das Bild ist der Input für eine Convolutional-Schicht mit  $5 \times 5$ -Kern (inkl. Bias), Schrittweite 1.

1. Das Output-Bild hat Breite  $W = 318$  und Höhe  $H = 238$ . **Wie groß ist das Padding der Schicht?**

# Lösung: CNN-Kanäle und Padding

## 1. **Padding:**

Output width =  $W - \text{Kernel} + 1 = 320 - 5 + 1 = 316$  (if no padding).  
Aber Output = 318, also wurde **Padding = 1** Pixel an jeder Seite verwendet.

# Kahoot!

Leer :(

## 5. Hausaufgaben

---

# Exercise 12: Intro ML III

Auf <https://expert.ethz.ch/enrolled/SS25/mavt2/exercises>

- K-means
- Grid Search for Polynomials
- Circles
- Dimensionality Reduction
- CNNs

Abgabedatum: Freitag 06.06.2025, 20:00 MEZ

## Freitag 15. August, 09:30 - 11:30

- Sucht euch eine Zusammenfassung aus und ergänzt diese von nun an (2 sind auf meiner Website)
- Versucht die Grundstrukturen von Coding Aufgaben zu verstehen und euch daran zu erinnern (Es kommt of das gleiche, aber nie exakt). Geht dafür auch alte Hausaufgaben durch
- Übt viel mit den Prüfungen (auch wenn ihr nur wenige habt). BAUG Prüfungen können ebenfalls nützlich sein, aber nicht alles darin ist relevant (werded ihr selber merken)
- Bei Unklarheiten: Schaut euch YT-Videos an, fragt das Internet / ChatGPT, fragt in den Foren auf moodle, **fragt nicht** (jederzeit, auch wenn ich vielleicht manchmal etwas länger brauche zum Antworten).

# Trauriges Goodbye

- Es hat mir sehr Spass gemacht die Übungsstunde zu leiten und es hat mich sehr gefreut immer wieder die gleichen Gesichter zu sehen.
- Ich hoffe Ihr konntet etwas lernen! (Vielleicht ja sogar von meinen Kommentaren auf CodeExpert...)
- Besucht mich doch nächstes Semester in meiner **Control Systems 1** Übungsstunde (bin mir aber noch nicht sicher)

# Feedback



<https://n.ethz.ch/~jschul/Feedback>