

Methods & Algorithms FS 2023

Aufgabenblatt 5: Tensorfaktorisierung

Die Bearbeitung der Aufgaben ist freiwillig; es erfolgt keine Bewertung.

Aufgabe 1

Gegeben sind folgende Matrizen

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}.$$

Berechnen Sie in Python das Kronecker-Produkt $\mathbf{A} \otimes \mathbf{B}$, das Khatri-Rao-Produkt $\mathbf{A} \odot \mathbf{B}$ und das Hadamard-Produkt $\mathbf{A} * \mathbf{B}$. Nutzen Sie das Modul `tensorly` für Ihre Berechnung.

Tipp: Erzeugen Sie sich die Matrizen mittels `np.arange()` und ändern Sie deren Gestalt mittels `reshape((#zeilen, #spalten))`.

Aufgabe 2

Erzeugen Sie sich einen 3-Mode-Tensor \mathcal{X} der Grösse $5 \times 2 \times 5$ mit den Zahlen von 1 bis 50.

```
x = np.arange(50).reshape((5,2,5))  
x = x + 1
```

Entfalten Sie den Tensor mithilfe des Pakets `tensorly` nach allen drei Moden.

Aufgabe 3

Erzeugen Sie sich einen 3-Mode-Tensor \mathcal{X} der Grösse $30 \times 30 \times 30$ bestehend aus Zufallszahlen.

```
x = np.random.rand(30,30,30)
```

Führen Sie sukzessive eine Tucker-Zerlegung von \mathcal{X} mit Rang $r \in [1, 50]$ gemäss

```
tucker(x, (r, r, r))
```

aus dem Paket `tensorly` durch, bestimmen Sie den jeweiligen Fehler des rekonstruierten Tensors und zeichnen Sie die Fehlerkurve. Was fällt auf?

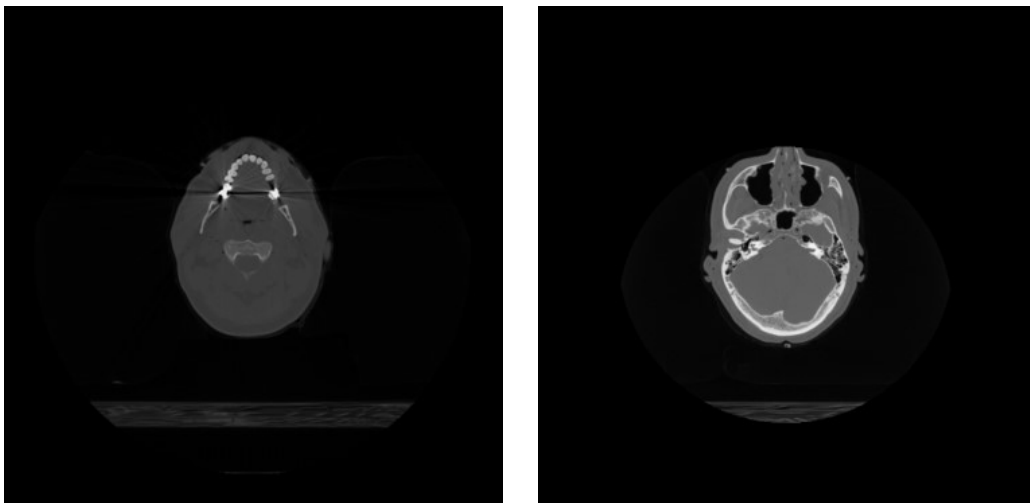
Aufgabe 4

Die Datei `MRI.npy` enthält einen Tensor der Grösse $245 \times 512 \times 512$ mit 245 MRI-Schnittbildern (Grösse jeweils 512×512 Pixel) eines menschlichen Kopfes. Laden und entpacken Sie zunächst die Datei `MRI.zip` aus Moodle, anschliessend können Sie den Tensor in Python folgendermassen einlesen.

```
import numpy as np  
  
X = np.load('MRI.npy')
```

Um einzelne Schnittbilder $i \in [0, 244]$ zu betrachten, können Sie diese mittels `imshow` aus dem Paket `matplotlib.pyplot` darstellen.

```
plt.imshow(X[i], cmap='gray')  
plt.show()
```



- Führen Sie eine Rang-1-Zerlegung mittels `parafac` aus dem Paket `tensorly` durch und stellen Sie das Ergebnis (analog zur Vorlesung) mit der Funktion `plot_uvw_one_feature` aus dem Paket `mytensor.py` dar. Welche Aussagen lassen sich über den Tensor treffen?
- Führen Sie eine Tensor-Zerlegung mittels `parafac` mit Rang $r \in \{1, 5, 10, 25, 40, 50\}$ durch, bestimmen Sie den jeweiligen Fehler des rekonstruierten Tensors und zeichnen Sie die Fehlerkurve. Was fällt auf? [**Achtung:** Die Zerlegungen sind sehr aufwändig und können insgesamt bis zu einer Stunde oder länger dauern.]
- Wie hoch ist die Kompressionsrate des Tensors bei einer Rang-50-Zerlegung gegenüber den Originaldaten?