

Modellierung WS 22-23 Blatt 1

Abgabe: 07.11.23 23:59

Auf diesem Übungsblatt schauen wir uns zum einen Numpy etwas genauer an. Ziel ist es hier die Dokumentation kennenzulernen und sich mit Grundlegenden operationen auseinander zusetzen.

Zum anderen betrachten wir wie Vektorräume Funktionen beschreiben können.

Aufgabe 1: Zuerst Numpy Basics

Bevor wir uns jedoch an eine richtige Aufgabe trauen, möchten wir ein besseres Gefühl für Numpy und seine Eigenarten entwickeln.

Das Ziel ist es sich hier insbesondere mit vektorisierter Rechnung auseinanderzusetzen, also **in keiner der Aufgaben Python- for Schleifen zu benutzen!**

Erstellen von Tensoren & einfache Rechnungen

1. Erstellen Sie einen Numpy-Tensor der ersten `1'000'000` Quadratzahlen.

Doku: [np.arange](#)

2. Erstellen Sie einen zufälligen Tensor der gleichen Größe mit zufälligen Fließkommazahlen zwischen `0` und `1` und sortieren Sie diesen.

Doku: [np.random](#) und [ndarray.sort](#)

Broadcasting

Doku: [Broadcasting](#)

(Ausgesprochen nützliches Werkzeug.)

1. Erstellen Sie eine Multiplikations/„Ein-Mal-Eins“ Tabelle mithilfe von **Broadcasting**.

Das Ergebnis sollte also so aussehen:

```
array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10],
       [ 2,  4,  6,  8, 10, 12, 14, 16, 18, 20],
       [ 3,  6,  9, 12, 15, 18, 21, 24, 27, 30],
       [ 4,  8, 12, 16, 20, 24, 28, 32, 36, 40],
       [ 5, 10, 15, 20, 25, 30, 35, 40, 45, 50],
       [ 6, 12, 18, 24, 30, 36, 42, 48, 54, 60],
       [ 7, 14, 21, 28, 35, 42, 49, 56, 63, 70],
       [ 8, 16, 24, 32, 40, 48, 56, 64, 72, 80],
       [ 9, 18, 27, 36, 45, 54, 63, 72, 81, 90],
       [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]])
```

2. Erstellen Sie einen zufälligen 2d-Tensor aus $\{0, 1\}^{1000 \times 1000}$ und teilen Sie alle Zahlen durch die jeweilige Spaltensumme.

Verifizieren Sie, dass die Spaltensumme des resultierenden Tensors stets 1 ist.

Reshaping & Indizierung

Doku: [Indizierung](#)

(Auch hier kann ich nur empfehlen die ganze Seite zu lesen).

1. Erstellen Sie eine 2d-Tensor (beispielsweise der Größe 10×10), der mit zufälligen ganzen Zahlen zwischen 1 und 10 gefüllt ist.

Geben Sie jeweils einen Ausdruck an, der

a. alle geraden Zahlen auf 0 setzt

b. den Anteil gerader Zahlen in einer beliebigen Matrix zählt Tipp: `.sum()`

direkt auf dem Bool-Tensor. Alle `true`-Werte werden als 1 gezählt, alle `false`-Werte als 0.

c. alle Spalten quadriert, die mindestens eine 5 enthalten

d. alle Diagonaleinträge auf 0 setzt.

e. die erste Spalte und die erste Zeile vertauscht

2. Erstellen Sie eine Funktion, die einen 2d-Tensor aus $\{0, 1\}^{n \times n}$ zurückgibt, der wie ein Schachbrett gefüllt ist.

Hinweis: `np.arange`, `np.reshape` und `%` (Modulo) Operation reichen hier aus.

Beispiel:

```

chess(9)
tensor([[1, 0, 1, 0, 1, 0, 1, 0, 1],
        [0, 1, 0, 1, 0, 1, 0, 1, 0],
        [1, 0, 1, 0, 1, 0, 1, 0, 1],
        [0, 1, 0, 1, 0, 1, 0, 1, 0],
        [1, 0, 1, 0, 1, 0, 1, 0, 1],
        [0, 1, 0, 1, 0, 1, 0, 1, 0],
        [1, 0, 1, 0, 1, 0, 1, 0, 1],
        [0, 1, 0, 1, 0, 1, 0, 1, 0],
        [1, 0, 1, 0, 1, 0, 1, 0, 1]], dtype=torch.int32)

```

3. Erstellen Sie eine Funktion, die einen „Hypercube“ der Kantenlänge `3` und Dimension `n` erstellt, bei dem jeweils überall eine `1` steht, außer im Zentrum, dort soll eine `0` stehen.
4. Etwas schwerer: Erstellen Sie eine Funktion, die einen „Hypercube“ der Kantenlänge `3` und Dimension `n` erstellt, bei dem jeweils eine `1` auf jeder Ecke steht, ansonsten aber nur mit `0` en gefüllt ist.

Hinweis: Das Schwierige ist hier insbesondere keine Schleife zu verwenden.

Konstruieren Sie zum Beispiel den Hypercube also für kleine Dimensionen per Hand, `flatten` (oder `.reshape(-1)` en) Sie den Würfel und prüfen Sie in welcher Frequenz sich die `1` en befinden.

Für die Lösung gibt es zwei Möglichkeiten:

- a. Testen Sie, ob die Zahlenfolge der Indices unter Der [Integer Sequence Database](#) finden.
- b. Nutzen Sie `np.meshgrid`.

Pi bestimmen + visualisieren.

Berechnen Sie näherungsweise π mit Hilfe von (Pseudo-)Zufallszahlen. Erzeugen Sie n zufällige Punkte innerhalb des Einheitsquadrates $[0, 1]^2$ und testen Sie ob diese innerhalb des Einheitskreises (mit Radius 1) liegen. Aus der Anzahl der Punkte die innerhalb des Viertelkreises (wir haben ja nur positive Punkte gewürfelt) liegen und der der Gesamtanzahl n lässt sich π berechnen. Dazu nehmen wir an, dass das Verhältnis der Punkte innerhalb des Viertelkreises und aller Punkte genau das Verhältnis zwischen den Flächeninhalten des Viertelkreises und der Quadrats ist.

$$\frac{\text{Anzahl Punkte im Kreis}}{\text{Anzahl Punkte im Einheitsquadrat}} \approx \frac{A_{\text{Viertelkreis mit Radius 1}}}{A_{\text{Einheitsquadrat}}} = \frac{\pi}{4}$$

1. Wählen Sie $n := 1$ Million und geben Sie den berechneten Wert für π wie beschrieben aus.
 2. Visualisieren Sie das Experiment (etwa mithilfe eines [Matplotlib Scatterplot](#)).
-

Aufgabe 2: Funktionenräume

Sei V der Vektorraum aller glatten (d.h. beliebig oft stetig differenzierbaren) Funktionen auf einem Intervall $[a, b] \in \mathbb{R}$.

1. Geben Sie Beispiele für Funktionen von V an. Welche Funktionen sind *nicht* in V enthalten?

Wir betrachten nun lineare Gleichungen

$$L(f) = g,$$

wobei $L : V \rightarrow V$ eine lineare Abbildung vom Funktionenraum V auf sich selbst ist.

2. Geben Sie ein Beispiel für eine Lineare Funktion L an.
3. Zeigen Sie (unter Benutzung der bekannten Eigenschaften von Ableitungen), dass die Abbildung $L = \frac{\partial^2}{\partial x^2}$ (doppeltes Ableiten einer Funktion) eine lineare Abbildung ist.
4. Geben Sie mindestens zwei nicht linear abhängige Funktionen an, die im Kern von $\frac{\partial^2}{\partial x^2}$ liegen.
5. Betrachten Sie die Differentialgleichung

$$a(x) \cdot \frac{\partial^2}{\partial x^2} f(x) + b(x) \cdot \frac{\partial}{\partial x} f(x) + c(x) \cdot f(x) + d(x) = 0$$

für beliebige, fest gegebene Funktionen $a, b, c, d \in C^\infty$ und eine unbekannte (gesuchte) Funktion $f \in C^\infty$ **Begründen Sie, warum die Menge aller Lösungen dieser Differentialgleichung einen affinen (Unter-)Vektorraum bildet.**