פעילות 3 - עבודה עם מטריצות תוך שימוש ב- NumPy

בפעילות זה נתרגל כתיבת קוד בשפת python לביצוע פעולות מתמטיות על מערכי מטריצות מסוג NumPy Arrays.

מקורות:

https://www.programiz.com/python-programming/matrix

מטריצה היא מערך דו מימדי של ערכים. מטריצה מאפשרת לנו לרכז בצורה יעלה מספר רב של ערכים כאשר לכל מטריצה קיים מכנה משותף לכל הערכים שבו. במערכות מכונה לומדת משתמשים במטריצות כדי לשמור את ערכי אותות המבוא והמוצא של רשתות נוירונים כמו כן את מערכי המשקלים של פרספטרונים המרכיבים את אבני הבסיס של כל רשת נוירונים.

בפעילות זו נלמד כיצד לקחת מטריצות במבנה numpy array ולבצע עליהם פעולות מתמטיות כמו חיבור וכפל.

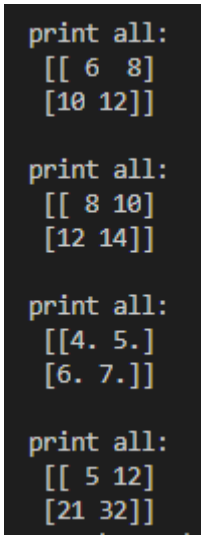
פעולות על מטריצות סימטריות

מטריצה ריבועית היא מטריצה שמספר העמודות שלה שווה למספר השורות.

נכתוב את הקוד הבא:

```
import numpy as np
a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])
x = a + b
print("\n print all:\n",x)
x=x+2
print("\n print all:\n",x)
x=x/2
print("\n print all:\n",x)
x=a*b
print("\n print all:\n",x)
```

נקבל את הפלט הבא:



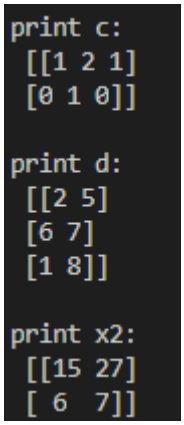
כפל מטריצות לא סימטריות

נכתוב את הקוד הבא:

```
import numpy as np
c = np.array([[1, 2, 1], [0, 1, 0]])
d = np.array([[2, 5], [6, 7], [1, 8]])

#x2=c*d  ValueError: operands could not be broadcast together with shapes (2,3) (3,2)
x2 = c.dot(d)
print("\nprint c:\n",c)
print("\nprint d:\n",d)
print("\nprint x2:\n",x2)
```

נקבל את הפלט הבא:



נבדוק את התוצאה:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 2 & 5 \\ 6 & 7 \\ 1 & 8 \end{bmatrix} = \begin{bmatrix} 15 & 27 \\ 6 & 7 \end{bmatrix}$$

כמובן שלא ניתן להכפיל כל מטריצה אחת בשנייה. עזרה בנושא ניתן לקבל באתר הבא:

Matrix Multiplication

http://matrixmultiplication.xyz/

שחלוף (transpose) מטריצות

שחלוף Transpose מטריצות היא פעולה מתמטית המחליפה בין השורות והעמודות של מטריצה נתונה.

נכתוב את הקוד הבא:

```
import numpy as np
x2 = np.array([[1, 2, 3, 4, 5],[6, 7, 8, 9, 10]])
a = x2.transpose()
b = x2.T
print("\n print x2:\n",x2)
print("\n print a:\n",a)
print("\n print b:\n",b)
```

נקבל את הפלט הבא:

```
print x2:
 [[ 1  2  3  4  5]
 [ 6  7  8  9 10]]

print a:
 [[ 1  6]
 [ 2  7]
 [ 3  8]
 [ 4  9]
 [ 5 10]]

print b:
 [[ 1  6]
 [ 2  7]
 [ 3  8]
 [ 4  9]
 [ 5 10]]
```

כפי שניתן לראות אין הבדל בין שימוש בפעולה transpose לבין שימוש ב- T כדי לבצע היפוך מטריצות

שימו לב, לא ניתן לבצע היפוך על מטריצה חד מימדית. אבל בהחלט ניתן לבצע היפוך על מטריצה דו מימדית בגודל של 1 על X (גודל כלשהו)

נסביר את הבעייה על ידי הדוגמה הבא:

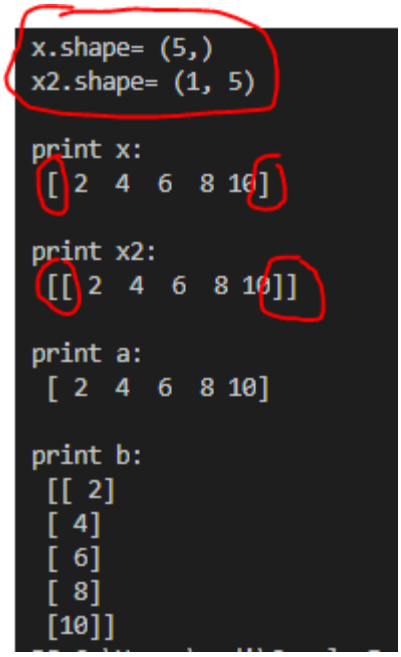
```
import numpy as np
x = np.array([2, 4, 6, 8, 10])
x2 = np.array([[2, 4, 6, 8, 10]])

a = x.T
b = x2.T

print("x.shape=",x.shape)
print("x2.shape=",x2.shape)

print("\n print x:\n",x)
print("\n print x2:\n",x2)
print("\n print a:\n",a)
print("\n print b:\n",b)
```

נקבל את הפלט הבא:



כדי להבין את העניין נתבונן בצורות המטריצות x ו- x2

מטריצה x היא חד מימדית:

x.shape= (5,)

מטריצה x2 היא דו מימדית:

x2.shape= (1, 5)

שימוש לב להגדרות השונות של 2 המטריצות:

```
x = np.array([2, 4, 6, 8, 10])
x2 = np.array([[2, 4, 6, 8, 10]])
```

תנאי השימוש

תנאי השימוש במסמך זה הם לפי הסטנדרט הבא:

You are free:

to Share – to copy, distribute and transmit the material
to Remix – to adapt the material

Under the following conditions:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
NonCommercial — You may not use the material for commercial purposes.
ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.