

Numpy Her Bir Elemana İşlem Yaptırmak

NumPy'da bir dizi (array) içerisindeki her elemana toplu bir şekilde işlem yapmak oldukça kolaydır ve NumPy'nin en güçlü özelliklerinden biridir. Buna **vektörel işlemler** (vectorized operations) veya **eleman bazında işlemler** (element-wise operations) denir. Python'daki standart döngüler (for loop) kullanmak yerine, işlemleri doğrudan dizi üzerinde gerçekleştirebilirsiniz. Bu hem daha okunaklı kod yazmanızı sağlar hem de işlemler C dilinde optimize edildiği için çok daha hızlı çalışır.

İşte birkaç yaygın yöntem:

- 1. Temel Aritmetik Operatörler:** Standart matematiksel operatörleri (+, -, *, /, //, %, **) doğrudan NumPy dizileriyle kullanabilirsiniz.
 - Bir skaler (tek bir sayı) ile işlem yapma: Operatör, dizinin her bir elemanına ayrı ayrı uygulanır.
 - Aynı boyuttaki iki dizi arasında işlem yapma: İşlem, karşılıklı gelen elemanlar arasında yapılır.
- 2. NumPy Evrensel Fonksiyonları (Universal Functions - ufuncs):** NumPy, matematiksel (`np.sqrt` , `np.sin` , `np.cos` , `np.log` , `np.exp` vb.), karşılaştırma (`np.greater` , `np.less` vb.) ve daha birçok işlemi eleman bazında yapan geniş bir fonksiyon kütüphanesine sahiptir.

Örnekler:

Python

```
import numpy as np
```

```
# Örnek bir NumPy dizisi oluşturalım
dizi = np.array([1, 2, 3, 4, 5])
print(f"Orijinal Dizi:\n{dizi}\n")
```

```
# 1. Skaler Değer ile Toplama (Her elemana 10 ekleme)
toplanmis_dizi = dizi + 10
print(f"Her elemana 10 eklendi:\n{toplanmis_dizi}\n")
```

```
# 2. Skaler Değer ile Çarpma (Her elemanı 3 ile çarpma)
carpilmis_dizi = dizi * 3
print(f"Her eleman 3 ile çarpıldı:\n{carpilmis_dizi}\n")
```

```
# 3. Elemanların Karesini Alma
```

```
kareleri_alinmis_dizi = dizi ** 2
# Alternatif olarak NumPy fonksiyonu kullanılabilir: kareleri_alinmis_dizi = np.square(dizi)
print(f"Her elemanın karesi:\n{kareleri_alinmis_dizi}\n")
```

```
# 4. NumPy Fonksiyonu Kullanma (Her elemanın karekökünü alma)
# Karekök için float dizi daha uygun olabilir veya sonuç float olur
kokleri_alinmis_dizi = np.sqrt(dizi)
print(f"Her elemanın karekökü:\n{kokleri_alinmis_dizi}\n")
```

```
# 5. İki Diziyi Eleman Bazında Toplama
dizi1 = np.array([10, 20, 30, 40, 50])
dizi2 = np.array([1, 2, 3, 4, 5])
```

```
toplam_diziler = dizi1 + dizi2
print(f"İki dizinin eleman bazında toplamı:\n{toplam_diziler}\n")
```

```
# 6. Koşullu İşlemler (Boolean Indexing ile birlikte)
# Örneğin, 3'ten büyük elemanları 10 ile çarpma
dizi_kopya = dizi.copy() # Orijinal diziyi değiştirmemek için kopyasını alalım
buyuk_olanlar_maskesi = dizi_kopya > 3
dizi_kopya[buyuk_olanlar_maskesi] = dizi_kopya[buyuk_olanlar_maskesi] * 10
# Alternatif ve daha kısa yol: dizi_kopya[dizi_kopya > 3] *= 10
print(f"3'ten büyük elemanlar 10 ile çarpıldı:\n{dizi_kopya}\n")
```

Kısacası, NumPy dizileriyle çalışırken eleman bazında işlem yapmak için genellikle Python döngüleri yazmanıza gerek yoktur. Doğrudan operatörleri veya NumPy fonksiyonlarını diziye uygulamanız yeterlidir. Bu, NumPy'nin temel avantajlarından biridir.