

YAPAY ZEKA VE PYTHON PROGRAMLAMA

Numpy
Emir Öztürk

Dr. Öğr. Üyesi Altan MESUT'un "Programlama Dillerine Giriş" ders slaytları kaynak alınmıştır.

NUMPY

- Python'da bilimsel işlemler yapmak için
- Dizi ve dizi fonksiyonları
- Dizilerin belirli türde olması
- pip3 install numpy

NumPy, Python'da bilimsel işlemler için kullanılan bir kütüphanedir. Yüksek performanslı çok boyutlu bir dizi nesnesi ve bu dizilerle çalışmak için araçlar sağlar. MATLAB'a benzer bir hesaplama yapısı vardır. Dizilerin listelerden farkı elemanların hepsinin belirli bir veri türünde olmasıdır. C dilindeki gibi sayı (int yada float) olarak saklanması hızlı işlem yapabilmesini sağlar.

NUMPY'DA DİZİLER

- Array()
- numpy.array([[1,2],[3,4]])
- Görüntüleme farkı
- type() farkı
- Listeleri içeren listelerde boyut eşitliği

NumPy'da diziler array yöntemi ile tanımlanır. Diziler listelerden farklı olarak ekranda satır satır görüntülenir. Numpy'da eğer birden fazla liste içeren listeler varsa alt liste boyutlarının eşit olması istenir.

BASİT VERİ TÜRÜ BELİRLEME

- Otomatik tür belirlenmesi
- Belirli türe zorlama
- dtype

Eğer dizi tanımlanırken belirli bir veri türüne zorlamak istenirse dtype parametresi kullanılabilir.

Dizinin türü type ile kontrol edilebilir, dizi tanımlanırken dtype ile tür verilebilir.

ÖZEL DİZİLER

- zeros()
- ones()
- full()
- eye()

zeros verilen boyut kadar bir 0 matrisi verir

ones verilen boyut kadar bir 1 matrisi verir

full verilen boyut ve verilen değer için o değerle dolu bir matris üretir

eye ise verilen boyut kadar birim matris elde edilmesini sağlar.

DİZİLERE RASTGELE ELEMAN ATAMA

- np.random.rand()
- np.random.randint()

Dizilerin rastgele atanmış değerler ile oluşturulması sağlanabilir.

Eğer 0 ile 1 aralığında float değil de belirli bir aralıkta int türünde rastgele değerler istenirse randint kullanılabilir. İlk parametre başlangıç, ikinci parametre bitiş ve 3. parametre boyut alır. Boyut tek bir int verilip tek boyutlu bir dizi elde edilebileceği gibi size =(n,n) verilip bir matris de elde edilebilir.

DİZİLERLE MANTIKSAL İŞLEMLER

- Skalar ile işlem
- Eleman bazlı
- bool kontroller

Diziler üzerinde skalarlar ile aritmetik işlemler gerçekleştirilebilir. Bu işlemler her eleman üzerine uygulanır. bool kontroller de yine diziler üzerinde gerçekleştirilebilmektedir.

DİZİLERLE MATEMATİKSEL İŞLEMLER

- Python listelerinde kullanılamaz
- +, *, **, -, >, <
- Dizi elemanlarının tümü için

MATRİS ÇARPIMI

- Matrislerin çarpımı
 - *
 - @
 - `matris.dot(matris2)`
 - `np.dot(matris1,matris2)`

“*” element çarpımı yaparken @ ise matris çarpımı gerçekleştirir. dot işlemi ise daha önceki sürümlerde @ işlemine karşılık gelmektedir. Array nesnesi üzerinde matris çarpımı için kullanılan @ operatörü Python 3.5 ile birlikte gelmiştir. Bu işlemi gerçekleştirmek için eski sürümlerde dot veya matmul yöntemleri kullanılmaktadır. Parametre olarak liste veya demet gibi yapıları da kabul ettiği için bu yöntemler halen kullanılmaktadır.

MATRIX NESNESİ

- İki boyutlu
- Array'e benzer
- * işlemi array'e göre farklı
- np.matrix

Array nesnesinden farklı olarak sadece 2 boyutlu olabilen bir nesnedir. Array nesnesine benzer şekilde veya bir string içinde satırları ayıran ; kullanımı tanımlanabilir.

Array üzerinde * operatörü eleman bazlı çarpma yaparken (eş indisli elemanları çarpar, yani + ile benzer), matrix üzerinde bu operatör matris çarpımı yapar. Yeni numpy sürümlerinde matrix nesnesi kullanılamamaktadır. Bunun yerine bir önceki slaytta verilen çok boyutlu diziler matris olarak işlenmektedir.

DİZİ FONKSİYONLARI

- sum(x)
- prod(x)
- mean(x)
- std(x)
- max(x)
- min(x)
- argmax(x)
- argmin(x)

np.sum(x): x dizisinin tüm elemanlarının toplamını verir.

np.prod(x): x dizisinin tüm elemanlarının çarpımını verir.

np.mean(x): x dizisinin tüm elemanlarının ortalamasını verir.

np.std(x): x dizisinin tüm elemanlarının standart sapmasını verir.

np.max(x): x dizisindeki en büyük elemanı verir

np.min(x): x dizisindeki en küçük elemanı verir

np.argmax(x): x dizisindeki en büyük elemanın indisini verir

np.argmin(x): x dizisindeki en küçük elemanın indisini verir

DİZİ FONKSİYONLARI

- all(x)
- any(x)
- sort(x)
- transpose(x)

np.all(x): x dizisinde tüm elemanlar sıfırdan farklıysa (boolean için hepsi True ise) True döndürür.

np.any(x): x dizisindeki elemanların biri bile sıfırdan farklıysa (boolean için biri True ise) True döndürür.

np.sort(x): x dizisindeki elemanları sıralayıp yine aynı diziye kopyalar.

Eğer axis=0 ikinci parametre olarak verilirse sütun bazında sıralar, aksi halde axis=1 kabul edilir ve satır bazlı sıralar.

np.transpose(x): x dizisinin transpozisini (devriğini (xT): yani satır-sütun yer değiştirmiş halini) verir. Dizi tek boyutlu ise, yani matris değilse herhangi bir

işlem yapmaz.

ÖRNEK

- Bir dizinin ortalamasından daha büyük olan elemanlarını yeni bir dizi olarak elde eden uygulama

```
for eleman in dizi:
    print(eleman)
    if eleman > ortalama:
        yeniDizi+=[eleman]
print(yeniDizi)
ya da
print(dizi[dizi>ortalama])
```

DİZİNİN BOYUTLARINI DEĞİŞTİRME

- Boyut arttırma
- Boyut azaltma
- reshape()

Bir dizinin boyutunu değiştirmek için reshape fonksiyonu kullanılabilir. Reshape ile verilen parametrelerin toplam boyutla aynı olması gerekir. 12'lik bir dizi (3,4) ya da (4,3) şeklinde yeniden boyutlandırılabilir ama (3,3) yapılamaz. Ayrıca yeniden tek boyuta düşürme işlemi için (1,12) kullanılabilir fakat bu durumda dizi içerisinde 1 dizi daha olacaktır. 12 elemanlı bir dizi oluşturulması isteniyorsa bu durumda (12,) kullanılmalıdır. (reshape(1,12) vs reshape(12,) farklıdır)

POLİNOMLAR

- `np.poly1d()`
- Dizi parametresi
- Katsayılar için
- `print` ile kullanılabilir

NumPy’de polinomlar ile çalışmayı sağlayan `poly1d` sınıfı bulunur.

Bu sınıf parametre olarak parantez içerisinde katsayı değerlerini alır. Örn. (1,2,1) verildiğinde x^2+2x+1 polinomu oluşturur.

POLİNOMLAR

- Polinomun bir noktadaki değeri elde edilebilir
- `polinom(değer)`
- Katsayılar için `polinom[indis]`

Tanımlanan bir polinomun n noktasındaki değerini bulmak için `polinom(n)` yazılabilir.

Bu polinomda `polinom[0]` en düşük dereceli katsayıdır, `p[1]` x ’in, `p[2]` ise x^2 ’nin katsayılarıdır. Yani indisleme polinomun tanım sırasının tersinde gider (x^n değeri için `p[n]`).

POLİNOM FONKSİYONLARI

- `order`
- `roots()`
- `deriv()`
- `integ()`

Bir polinomun derecesi `order` ile öğrenilebilir. Köklerini bulmak için ise `roots` metodu kullanılabilir.

Bir polinomun türevi için `deriv()`, integrali için `integ()` yöntemleri kullanılabilir.