دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی برق

گزارش آزمایش اول
مقدمه‌ای بر هوش محاسباتی
دکتر عبداللهی

نام و نام خانوادگی
کیان بهزاد

شماره دانشجویی
۹۵۲۳۰۱۷

پیاده سازی تابع در محیط های پایتون و متلب

در ابتدا قصد داریم به پیاده سازی تابع در **پایتون** بپردازیم
به منظور تعریف تابع در پایتون در ابتدا از کلمه کلیدی def استفاده می شود، سپس می‌بایست نام تابع را مشخص کنیم و درون پرانتز ورودی های آن را تعیین می‌کنیم.

```
def my_function():
  print("Hello from a function")
```

در صورت استفاده از return، تابع خروجی خواهد داشت و در غیر این صورت به صورت void خواهد بود.

```
def my_function(x):
  return 5 * x
```

حال در قسمت اول این آزمایش می‌خواهیم تابع sigmoid را پیاده کنیم.
این کار به صورت روبرو انجام می‌شود.

```
def sigmoid(x: np.ndarray):
    if type(x) == np.ndarray:
        a = np.arange(x.shape[0] * x.shape[1], dtype=np.float).reshape(x.shape[0], x.shape[1])
        for i in range(a.shape[0]):
            for j in range(a.shape[1]):
                a[i][j] = sigmoid(x[i][j])
        return a

    if type(x) == np.int64 or type(x) == np.float:
        return (1 / (1 + math.exp(-x)))
```

همانطور که پیداست این تابع یک ورودی دارد که نوع آن در حالت general یک nparray است.
اما با در نظر گرفتن if ها در تابع تعیین می‌شود نوع درودی چیست.
اگر float بود خروجی نیز مقدار sigmoid آن است و در غیر این صورت یک ماتریس را که از تک تک المان هایش sigmoid گرفته شده بر می‌گردانیم.

حال به پیاده سازی تابع در **متلب** نگاهی می‌اندازیم.
به منظور تعریف تابع در پایتون در ابتدا از کلمه کلیدی function استفاده می شود، سپس می‌بایست نام خروجی های تابع را مشخص کنیم، سپس نوبت به نام و ورودی های تابع می‌رسد.

```
function ave = average(x)
    ave = sum(x(:))/numel(x);
end
```

در متلب می‌توان همانند زیر چندین خروجی تعریف کرد.

```
function [m,s] = stat(x)
    n = length(x);
    m = sum(x)/n;
    s = sqrt(sum((x-m).^2/n));
end
```

حال در این آزمایش می‌خواهیم تابع sigmoid را پیاده کنیم.
این کار به صورت روبرو انجام می‌شود.

```
function sig = sigmoidMatrix(a)
    mySize = size(a);
    sig = zeros(mySize(1), mySize(2));
    d = mySize(1)+mySize(2) + 1;
    for i = 1:d
        sig(i) = sigmoid(a(i));
    end
end
```

همانطور که پیداست این تابع یک ماتریس می‌گیرد و sigmoid تک تک المان های آن را بر کی‌گرداند.

تمرین

تابعی بنویسید که یک ماتریس و ابعاد آن را از ورودی دریافت کند و $\sum_{i=1}^{n}\sum_{j=1}^{m} tanh(A_{ij})$ را باز گرداند.(A نمایش دهنده ماتریس ورودی تابع و mو n تعداد سطرها و ستون های آن هستند) تابع $tanh(x)$ به صورت زیر تعریف می شود.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad (۳.۱)$$

```
function res = tanh(x)
    res = (exp(x) - exp(-x)) / (exp(x) + exp(-x));
end

function [sumTanh] = tanhmatrix(a)
    mySize = size(a);
    d = mySize(1)+mySize(2) + 1;
    sumTanh = 0;
    for i = 1:d
        sumTanh = sumTanh + tanh(a(i));
    end

end
```

```
import math
import numpy as np

def tanh(x: float):
    return (math.exp(x) - math.exp(-x)) / (math.exp(x) + math.exp(-x))

def tanhmatrix(a: np.ndarray):
    sumTanh = 0
    for i in np.nditer(a):
        sumTanh += tanh(i)
    return sumTanh

a = np.array([[1, 2, 3], [1, 2, 3]])
print(tanhmatrix(a))
```

گزارش و کد را می‌توانید در repository زیر ببینید(Az_jalase1)

https://github.com/kianbehzad/computational-intelligence