

## پروژه اول - مقدار دهی اولیه

آموزش شبکه عصبی نیاز به تعیین مقدار اولیه وزن ها دارد. یک روش مقدار دهی به سرعت آموزش کمک میکند.

در اینجا بررسی میکنیم که چگونه می توان مقادیر اولیه را برای یک شبکه عصبی انتخاب کرد؟ در این پروژه، خواهید دید که چگونه مقادیر اولیه متفاوت منجر به نتایج متفاوت می شود.

یک مقدار دهی اولیه خوب می تواند:

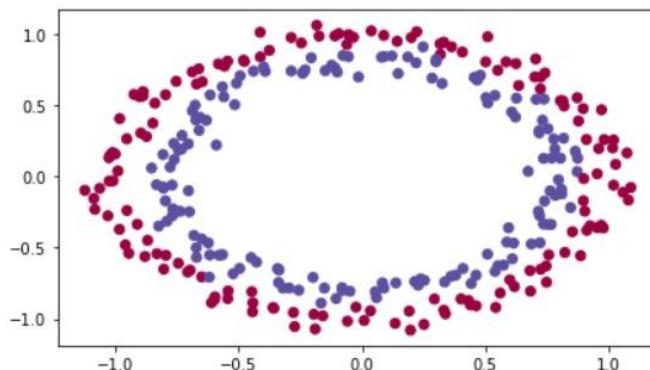
- سرعت همگرایی شبکه را افزایش دهد.
- فرست نزول گرادیان را افزایش دهد و به خطای آموزش پایین تر (و تعمیم پذیری بیشتر) نزدیک شود.

### مرحله صفر: اضافه کردن کتابخانه های لازم

فایل `init_utils.py` موجود در پوشه پروژه را در مسیر پروژه قرار دهید.

```
import math
import numpy as np
import matplotlib.pyplot as plt
import sklearn
import sklearn.datasets
from init_utils import sigmoid, relu, compute_loss, forward_propagation, backward_propagation
from init_utils import update_parameters, predict, load_dataset, plot_decision_boundary, predict_dec
```

مشاهده میکنید که داده ها پس از اجرای سلول اول نمایش داده می شوند.



### مرحله اول: مدل شبکه عصبی

شبکه عصبی سه لایه در این مرحله مدلسازی می شود. مقدار دهی اولیه در سه حالت زیر انجام می شود:

- مقدار دهی با صفر
- مقدار دهی تصادفی
- مقدار دهی He

## مرحله دوم: مقدار دهی صفر

در شبکه های عصبی دو دسته پارامتر باید مقدار دهی اولیه داشته باشند:

- ماتریس های وزن
- بردارهای بایاس

تمرین اول: در این بخش بردار وزن و بایاس را با صفر مقدار دهی کنید. برای محاسبه ابعاد ماتریس ها از مرحله قبل و کد مدلسازی شبکه عصبی استفاده کنید.

پس از مقدار دهی نتیجه آموزش شبکه را به ازای این مقدار دهی مشاهده کنید و عملکرد آن را بررسی کنید.

به خاطر داشته باشید که وزن ها بهتر است به نحوی وزن دهی شوند که تقارن آن ها شکسته شود. مقدار دهی اولیه بایاس ها با مقدار صفر مشکلی ایجاد نمیکند.

## مرحله سوم: مقدار دهی تصادفی

برای شکستن تقارن وزن ها را به طور تصادفی مقدار دهی می کنیم. پس از مقدار دهی اولیه ، هر نورون می تواند عملکرد متفاوتی از ورودی های خود را بیاموزد.

تمرین دوم: مقدار دهی اولیه بایاس را میتوانید با مقدار دهی صفر انجام دهید اما مقدار دهی وزن های شبکه را به صورت تصادفی پیاده سازی کنید و عملکرد شبکه را بررسی کنید. (می توانید از تابع `np.random.randn() * 10` استفاده کنید.)

## مرحله چهارم: مقدار دهی He

مقدار دهی اولیه He شبیه به مقدار دهی اولیه Xavier عمل میکند. رابطه مقدار دهی He به صورت زیر است:

$$\sqrt{\frac{2}{\text{dimension of the previous layer}}}$$

تمرین سوم: وزن ها را با مقدار دهی He به صورت اولیه مقدار دهی کنید و عملکرد شبکه را بررسی کنید. (این مقدار دهی مانند مقدار تصادفی از تابع `randn` استفاده میکند اما به جای این که مقدار تصادفی را در  $10$  ضرب کنیم، مقدار آن را در رابطه He ضرب میکنیم.)

در مورد تفاوت ضرب مقدار دهی تصادفی در  $10$  و ضرب آن در رابطه He تحقیق کنید و عملکرد شبکه را مشاهده کنید. کدامیک بهتر عمل میکند؟