



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

استاد درس: دکتر فاطمه شاکری

زمستان ۱۴۰۲

پیاده سازی Perceptron

درس مباحثی در علوم کامپیوتر



نحوه پیاده سازی

۱. با استفاده از تابع `make_blobs` از کتابخانه `sklearn` به کمک قطعه کد زیر، مجموعه داده ای بسازید که به صورت خطی جدایی پذیر باشد. هر ستون از ماتریس X شامل ویژگی های یک نمونه داده است.

```
from sklearn.datasets import make_blobs

X, y = make_blobs(n_samples=100, n_features=10, centers=2, random_state=41)
```

۲. ابتدا کلاسی به نام F در نظر بگیرید و یکی از متدهای آن را به نام `sign` پیاده سازی کنید. متدهای این کلاس متعلق به کلاس هستند و باید به صورت `static` تعریف شوند. در حقیقت این کلاس مسئول نگهداری توابع کاربردی برای انجام محاسبات است که در اینجا ما فقط می خواهیم `sign` را تعریف کنیم.

```
class F:
    @staticmethod
    def sign(x):
        # Implement the sign activation function
        pass
```

۳. کلاس `Perceptron` را مانند تکه کد زیر در نظر بگیرید. این موجودیت تنها وظیفه نگهداری وزن های پرسپترون ما و محاسبه y براساس ورودی ما یعنی x را دارد. برای مقدار دهی اولیه وزن ها و بایاس می توانید از کتابخانه `numpy` استفاده کنید.

```
class Perceptron:
    def __init__(self, input_size):
        # Initialize the weights and biases of the new model
        pass

    def forward(self, input):
        # Feed input features to your model and return output
        pass
```



۴. حال کلاس Optimizer را مطابق قطعه کد زیر پیاده سازی کنید. این کلاس وظیفه این را دارد که مدل ما را بهینه کند. بدین معنی که در هر مرحله از یادگیری سعی کند وزن ها را در جهتی بروزرسانی کند که در نهایت ما هیچ خطایی نداشته باشیم (با فرض خطی جدایی پذیر بودن داده ها)

```
class Optimizer:
    def __init__(self, model):
        pass

    def update(self, x, y, y_hat):
        # Update weight and bias of the model
        pass
```

۵. با توجه به تعداد ویژگی های داده های موجود، یک شی Perceptron بسازید (این پارامتر باید با تعداد ویژگی های قسمت اول یکسان باشد) در مرحله بعدی به optimizer اجازه دهید تا به مدل شما دسترسی داشته باشد تا بتواند آن را بهینه کند.

۶. در این قسمت شما باید با پیاده سازی فرآیندی مناسب با استفاده از instance های optimizer و perceptron وضعیت مدل را بهبود بدهید تا به ابر صفحه جدا کننده نزدیک تر شود.

```
n_iter = 1000

for _ in range(n_iter):
    for x, y in zip(X, Y):
        # Complete Code
```

۷. به ازای n_features برابر ۲، نقاط مجموعه داده ی خود و خط حاصل از Perceptron خود را به کمک کتابخانه Matplotlib در فضای ۲ بعدی نمایش دهید.

معیارهای ارزیابی

- برای محاسبات شما مجاز هستید تنها از کتابخانه numpy استفاده کنید.
- از شما انتظار می رود که Perceptron خود را طبق مطالب تدریس شده در کلاس پیاده سازی کنید.
- از شما انتظار می رود در کلاس Optimizer بروزرسانی وزن ها را دقیقاً مطابق با آنچه در کلاس گفته شد پیاده سازی کنید.
- پیاده سازی کد تنها در ساختار گفته شده قابل قبول می باشد. به طور مثال خود مدل نباید وزن های خودش را آپدیت کند یا اینکه در حلقه یادگیری نباید شما مستقیم وزن ها را آپدیت کنید.



- پیاده سازی خارج از قالب گفته شده هیچ نمره ای ندارد

نحوه ارسال

یک فایل زیپ شامل فایل جویپتر نوت بوک^۱، فایل حاوی لینک گوگل کولب^۲ و در نهایت فایل داکيومنت پروژه را ارسال کنید. شما همچنین میتوانید داکيومنت خودتون رو درون جویپتر نوت بوک قرار دهید و فایل داکيومنت جدایی ارسال نکنید. تمامی گزارش ها باید به زبان فارسی باشند. دقت داشته باشید تنها فرستادن لینک گوگل کولب پذیرفته شده نیست. قاعده نامگذاری فایل زیپ:

project_n_studentNumber.zip

که n همان شماره پروژه و studentNumber شماره دانشجویی میباشد.

مهلت

برای ارسال پروژه سری ۱ تا ساعت ۲۳:۵۹ ۲۵ اسفند زمان دارید.

¹Jupyter Notebook

²Google Colab