

تشخیص اعداد زوج و فرد با استفاده از پرسپترون ها

می‌توانید با استفاده از پرسپترون‌های ساده، اعداد زوج و فرد را تشخیص دهید. برای این کار، ابتدا لازم است که یک پرسپترون ساده برای تشخیص اعداد زوج و فرد آموزش دهید.

در اینجا، ما از کتابخانه `numpy` برای ایجاد و مدیریت ماتریس‌ها و بردارها استفاده می‌کنیم. همچنین از کتابخانه `random` برای ایجاد اعداد تصادفی برای آموزش پرسپترون استفاده می‌کنیم.

ابتدا، وزن‌ها و بایاس پرسپترون را با استفاده از اعداد تصادفی ایجاد می‌کنیم. سپس با استفاده از تابع فعال‌سازی سیگموئید، خروجی پرسپترون را برای هر ورودی محاسبه می‌کنیم. در نهایت، با مقایسه خروجی پرسپترون با مقدار صفر، می‌توانیم تشخیص دهیم که آیا عدد ورودی زوج است یا خیر.

کد زیر این کار را انجام می‌دهد:

```
```python
import numpy as np

class Perceptron:
 def __init__(self, input_size):
 self.weights = np.random.rand(input_size)
 self.bias = np.random.rand()

 def predict(self, input):
 activation = np.dot(input, self.weights) + self.bias
 output = self.sigmoid(activation)
 return output

 def sigmoid(self, x):
 return 1 / (1 + np.exp(-x))

even_odd_classifier = Perceptron(1)

آموزش پرسپترون برای تشخیص اعداد زوج و فرد
for i in range(10000):
 # ایجاد یک عدد تصادفی
 n = np.random.randint(1000)
 # مقدار انتظار داده شده برای خروجی پرسپترون
 expected_output = 0 if n % 2 == 0 else 1
 # ورودی به پرسپترون
 input = np.array([n])
 # خروجی پرسپترون
 output = even_odd_classifier.predict(input)
 # خطای پرسپترون
 error = expected_output - output
 # بهروزرسانی وزن‌ها و بایاس
```

```
even_odd_classifier.weights += error * input
even_odd_classifier.bias += error
```

```
تست پرسپترون با اعداد زوج و فرد
for i in range(10):
 n = np.random.randint(1000)
 input = np.array([n])
 output = even_odd_classifier.predict(input)
 if output < 0.5:
 print(n, "is even")
 else:
 print(n, "is odd")
...
```

در این کد، ابتدا یک کلاس `Perceptron` ایجاد شده است که شامل توابع `\_\_init\_\_`، `predict` و `sigmoid` است. تابع `\_\_init\_\_` برای ایجاد وزن‌ها و بایاس پرسپترون استفاده می‌شود. تابع `predict` ورودی را به پرسپترون می‌دهد و خروجی را محاسبه می‌کند. تابع `sigmoid` نیز تابع فعال‌سازی سیگموئید را پیاده‌سازی می‌کند.

سپس یک نمونه از پرسپترون برای تشخیص اعداد زوج و فرد ایجاد شده است. پس از آن، پرسپترون با استفاده از یک حلقه آموزش، برای تشخیص اعداد زوج و فرد آموزش داده می‌شود. در هر مرحله از حلقه آموزش، یک عدد تصادفی ایجاد می‌شود و با استفاده از مقدار انتظار داده شده برای خروجی پرسپترون، خطای پرسپترون محاسبه شده و وزن‌ها و بایاس به‌روزرسانی می‌شوند.

در نهایت، پرسپترون با استفاده از چند عدد تصادفی برای تست، بررسی می‌شود که آیا هر عدد زوج است یا خیر.