

کلاس‌های ارائه شده در API

بیشتر کلاس‌هایی که در این پروژه استفاده شده‌اند از این دو کلاس **abstract** ارث برده‌اند:

ValuedElement

این کلاس به هر المان اجازه می‌دهد تا یک مقدار **set** شدن داشته باشد. به عنوان مثال کلاس‌های **Input** و **Weight** از این کلاس به ارث برده‌اند. زیرکلاس‌های این کلاس باید حتماً این سه تابع را پیاده‌سازی بکنند:

```
set_value(self, val)
get_value(self)
get_name(self)
```

DifferentiableElement

این کلاس برای المان‌هایی استفاده می‌شود که خروجی دارند و در فرآیند مشتق‌گیری نیز حضور دارند. این کلاس‌ها دو تابع زیر را دارند:

```
output(self)
dOutdX(self, elem)
```

کلاس‌های **Input**, **Neuron** و **PerformanceElem** همگی از این کلاس به ارث برده‌اند. شما باید این توابع را برای این ۳ کلاس پیاده‌سازی کنید.

Weight(ValuedElement)

این کلاس که زیرکلاس **ValuedElement** است، برای وزن‌های شبکه استفاده می‌شود و قابلیت آپدیت شدن دارد. این کلاس علاوه بر توابع کلاس پدرش توابع زیر را نیز دارد:

```
set_next_value(self, val):
```

مقدار فیلد **next_value** را تعیین می‌کند.

```
update(self):
```

مقدار فعلی وزن را برابر با فیلد **next_value** می‌شمارد و قرار می‌دهد.

Input(DifferentiableElement, ValuedElement)

این کلاس برای ورودی‌های شبکه استفاده می‌شود. این ورودی می‌تواند متغیر یا ثابت (ورودی‌های 1-ای که ضریب آن‌ها در واقع مقدار **bias** را نشان می‌دهد) باشد. دقت کنید که چون در تعیین مقدار ورودی‌ها وزنی دخیل نیست، تابع **dOutdX** این کلاس باید مقدار صفر برگرداند.

Neuron(DifferentiableElement)

این کلاس نماینده‌ی نورون‌های شبکه است. قسمتی از کد این کلاس توسط ما پیاده‌سازی شده است. به طور خاص، از یک مکانیزم **caching** استفاده کرده‌ایم که فرآیند **train** و **test** با سرعت بیشتر انجام شود. در نتیجه به جای پیاده‌سازی توابع **output** و **dOutdX** شما باید دو تابع **compute_output** و **compute_doutdx** را پیاده‌سازی کنید.

PerformanceElem(DifferentiableElement)

این کلاس نشانگر یک Performance Element است و به شما اجازه می‌دهد تا خروجی درست را با استفاده از تابع `set_desired` مشخص کنید.