

# آزمایش سوم (پیاده سازی شبکه عصبی چند لایه)

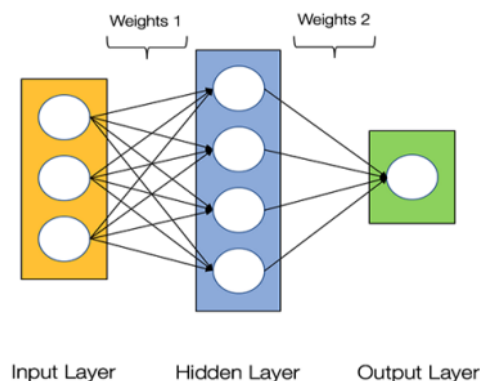
## ۱.۳ پیش گزارش

۱. محدودیت های مدل پرسپترون را بیان کنید.
۲. ساختار کلی یک شبکه عصبی ۲ لایه را رسم کنید
۳. مراحل الگوریتم پس انتشار خطا را با جزئیات بیان کنید.

## ۲.۳ مقدمه

مدل در نظر گرفته شده برای پرسپترون از محدودیت های زیادی رنج می برد برای مثال تنها توابع جداپذیر خطی امکان دسته بندی توسط این نوع مدل را دارند. مدل پرسپترون حتی توانایی پیش بینی تابع ساده  $XoR$  را ندارد. شبکه های عصبی چندلایه مدل توسعه یافته پرسپترون هستند و مشکلات مدل پرسپترون را ندارند. در این آزمایش به پیاده سازی یک شبکه عصبی ۲ لایه پرداخته می شود. ساختار کلی یک شبکه عصبی ۲ لایه در شکل ۱.۳ نمایش داده شده است. عناصر اصلی تشکیل دهنده شبکه عصبی ۲ لایه عبارتند از:

۱. لایه ورودی
۲. لایه مخفی
۳. لایه خروجی
۴. مجموعه ای از وزن ها و بایاس



شکل ۱.۳: ساختار کلی یک شبکه عصبی ۲ لایه

## ۵. توابع فعال ساز

انتخاب های مختلفی برای توابع فعال ساز یک شبکه عصبی وجود دارد. از توابع فعال ساز پرکاربرد می توان به  $\tanh$  و  $\text{sigmoid}$  اشاره کرد

خروجی یک شبکه عصبی ۲ لایه به صورت معادلات ۱.۳ است. در معادلات ۱.۳  $\sigma$  متناظر با تابع فعال ساز شبکه و  $W_1$  و  $W_2$  به ترتیب متناظر با وزن های لایه اول و دوم و  $b_1$  و  $b_2$  به ترتیب متناظر با بایاس لایه اول و دوم در نظر گرفته شده است.

$$\hat{y}(x) = \sigma(W_2 \sigma(w_1 x + b_1) + b_2) \quad (1.3)$$

برای سنجش عملکرد شبکه یک تابع هزینه در نظر گرفته می شود این تابع معمولاً به صورت توان ۲ خطا است و به صورت زیر تعریف می شود.

$$\text{cost} - \text{function} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.3)$$

هدف شبکه عصبی ۲ لایه پیدا کردن مقدار وزن ها و بایاس است به گونه ای که تابع هزینه در نظر گرفته شده به کمترین مقدار خود برسد. به پیدا کردن مقادیر مناسب وزن ها و بایاس های در شبکه عصبی برای بهبود عملکرد آن، یادگیری شبکه گفته می شود.

هر مرحله یادگیری شامل مراحل زیر است:

۱. محاسبه خروجی شبکه به ازای داده ورودی به این مرحله  $\text{forwardpass}$  گفته می شود

۲. تازه سازی وزن ها و بایاس های شبکه بر اساس خطا. به این مرحله پس انتشار خطا گفته می شود

در ادامه به مروری کوتاه بر روی الگوریتم پس انتشار خطا پرداخته می شود.

### ۱.۲.۳ الگوریتم پس انتشار خطا

مطابق با الگوریتم پس انتشار خطا برای پیدا کردن بهترین وزن ها که منجر به کمینه شدن تابع هزینه می شوند باید در خلاف جهت مشتق تابع هزینه بر حسب وزن ها حرکت کرد.

اگر تابع هزینه را برابر با توان ۲ خطا در نظر گرفته شود (معادله ۲.۳) ، به دلیل اینکه وزن ها در تابع هزینه به صورت مستقیم قرار ندارند ، نمی توان به صورت مستقیم از تابع هزینه بر حسب وزن مشتق گرفت . برای مشتق گرفتن از تابع هزینه بر حسب وزن می توان از قانون زنجیره ای استفاده کرد برای مثال برای تازه سازی وزن های لایه اول شبکه می توان به صورت زیر عمل کرد:

$$\begin{aligned} Loss(y, \hat{y}) &= \sum_{i=1}^n (y - \hat{y})^2 \\ \frac{\partial Loss(y, \hat{y})}{\partial W} &= \frac{\partial Loss(y, \hat{y})}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial z} * \frac{\partial z}{\partial W} \quad \text{where } z = Wx + b \\ &= 2(y - \hat{y}) * \text{derivative of sigmoid function} * x \\ &= 2(y - \hat{y}) * z(1-z) * x \end{aligned}$$

وزن های لایه دوم نیز به روش مشابه تازه سازی می شوند.

### ۳.۳ شرح آزمایش

در این قسمت به پیاده سازی یک شبکه عصبی ۲ لایه در محیط پایتون پرداخته می شود.

#### ۱.۳.۳ پیاده سازی شبکه عصبی ۲ لایه

برای پیاده سازی شبکه عصبی ۲ لایه یک کلاس به اسم *deep network* ایجاد کنید و مراحل زیر را دنبال کنید.

۱. یک تابع در کلاس بنویسید که مقدار دهی اولیه وزن ها و بایاس های لایه اول و دوم را انجام دهد (مقدار دهی اولیه را می توانید در تابع سازنده کلاس نیز انجام دهید)

۲. تابعی بنویسید که خروجی لایه اول شبکه را محاسبه کند (تابع *sigmoid* را به عنوان تابع فعالساز در نظر بگیرید)

۳. یک تابع برای محاسبه *sigmoid* پیاده سازی کنید

۴. یک تابع برای محاسبه مشتق تابع *sigmoid* پیاده سازی کنید.

۵. یک تابع برای محاسبه ضرب داخلی وزن ها و ورودی شبکه بنویسید

۶. با استفاده از توابعی که پیاده سازی کرده اید، تابعی بنویسید که وزن های لایه اول و دوم را تازه سازی کند.

### ۲.۳.۳ تست مدل

همان طور که می دانید تابع  $xor$  را نمی توان با یک پرسپترون ساده تخمین زد . در این قسمت برای ارزیابی مدل خود از تابع  $xor$  استفاده کنید.

ورودی و خروجی های تابع  $xor$  در شکل ۲.۳ نمایش داده شده است . قسمتی از داده را برای یادگیری و قسمتی دیگر را برای تست در نظر بگیرید. ورودی و خروجی ها را با استفاده از آرایه ها به مدل پیاده سازی شده خود دهید. دقت مدل پیاده سازی شده بر روی داده تست و یادگیری چه قدر است؟

X1	X2	X3	Y
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0

شکل ۲.۳: تابع  $xor$

### ۴.۳ تمرین

۱. مدل شبکه عصبی ۲ لایه پیاده سازی شده در محیط پایتون را دوباره در محیط متلب تکرار کنید.
۲. داده های فایل پیوست مربوط به یک ایستگاه آب و هوایی برای اندازه گیری آلودگی هوا در اختیار شما قرار داده شده است.

(<https://drive.google.com/file/d/1mlUIVKHWoQmviOIQ9qmcLtY6zy0eSz0/view?usp=sharing>)

- (آ) دسته بندی مناسب و پارامترهای مناسب برای دسته بندی داده ها را پیدا کنید.
- (ب) سعی کنید با نرم افزار متلب نیز شبکه بیش از ۳ لایه پیاده سازی کرده و نتایج را بررسی کنید. تاثیر افزایش لایه ها را بررسی کنید.
- (ج) تا چه حد میتوان این داده ها را بهبود داد.
- (د)  $Overfit$  را بررسی کرده و تغییرات داده های ولیدیشن و تست را نسبت به خطای داده ها به دست آورید.
- (ه) افزایش لایه ها در شبکه چه تاثیری میگذارد ؟
- (و) امتیازی: سعی کنید از فایل زیر تشخیص دهید کدام ایستگاه ها به ایستگاه ۱ نزدیک ترند.

(<https://drive.google.com/file/d/1WP7sRG01yq1V6cGOQVPlr4y152wruJN/view?usp=sharing>)