تکلیف نهم درس شناسایی الگو

وحید ملکی
شماره دانشجویی: ۴۰۳۱۳۰۰۴

۲۹ آذر ۱۴۰۴

سؤال ۱۵ (پایان ترم ۱۴۰۳)

تابع فعالیت (Activation Function) به صورت زیر داده شده است (تابع پله‌ای):

$$\phi(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{O.W.} \end{cases}$$

الف) پیاده‌سازی تابع **AND** با n ورودی

تابع AND تنها زمانی خروجی ۱ می‌دهد که **تمام** ورودی‌ها برابر با ۱ باشند. اگر حتی یکی از ورودی‌ها ۰ باشد، خروجی باید ۰ شود.

برای پیاده‌سازی این تابع با یک پرسپترون تک‌لایه:

- وزن تمام ورودی‌ها را برابر با ۱ در نظر می‌گیریم: $w_1 = w_2 = \cdots = w_n = 1$.
- ورودی خالص (Net Input) برابر است با مجموع ورودی‌ها به اضافه بایاس: $z = \sum_{i=1}^{n} x_i + b$.

شرایط مرزی:

۱. اگر تمام ورودی‌ها ۱ باشند (مجموع برابر n): خروجی باید ۱ باشد.

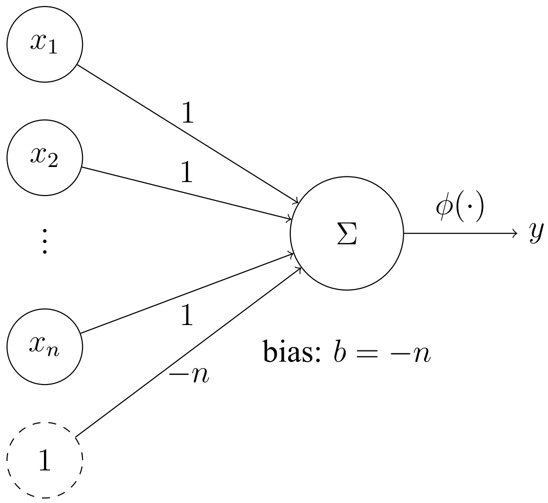
$$n + b \geq 0 \implies b \geq -n$$

۲. اگر $n-1$ ورودی ۱ باشند و یکی ۰ باشد (مجموع برابر $n-1$): خروجی باید ۰ باشد.

$$(n-1) + b < 0 \implies b < -(n-1) \implies b < 1 - n$$

با انتخاب $b = -n$ شرط اول ($n - n = 0 \geq 0$) برقرار می‌شود و شرط دوم ($n - 1 - n = -1 < 0$) نیز ارضا می‌گردد. بنابراین پارامترها عبارتند از:

$$w_i = 1, \quad b = -n$$



شکل ۱: شبکه عصبی تک لایه برای تابع AND

ب) پیاده‌سازی تابع OR با n ورودی

تابع OR زمانی خروجی ۱ می‌دهد که **حداقل یک** ورودی برابر با ۱ باشد. تنها زمانی خروجی ۰ است که تمام ورودی‌ها ۰ باشند.

مشابه حالت قبل، وزن‌ها را ۱ در نظر می‌گیریم ($w_i = 1$). شرایط مرزی:

۱. اگر تمام ورودی‌ها ۰ باشند (مجموع ۰): خروجی باید ۰ باشد.

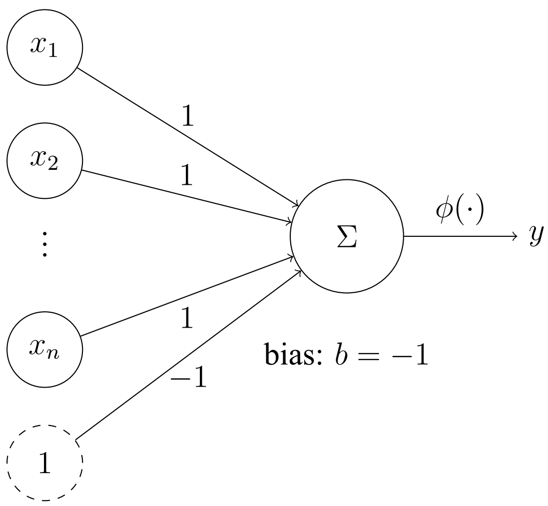
$$0 + b < 0 \implies b < 0$$

۲. اگر حداقل یک ورودی ۱ باشد (مجموع ≥ 1): خروجی باید ۱ باشد.

$$1 + b \geq 0 \implies b \geq -1$$

با انتخاب $b = -1$ شرط دوم ($1 - 1 = 0 \geq 0$) برقرار می‌شود و شرط اول ($0 - 1 = -1 < 0$) نیز صحیح است. (همچنین می‌توان $b = -0.5$ را نیز انتخاب کرد). بنابراین پارامترها:

$$w_i = 1, \quad b = -1$$



شکل ۲: شبکه عصبی تک لایه برای تابع OR

ج) عدم امکان پیاده‌سازی XOR با کمتر از دو لایه

تابع XOR (یا "یا انحصاری") برای دو ورودی باینری به صورت زیر تعریف می‌شود:

- $(0,0) \to 0$
- $(1,1) \to 0$
- $(0,1) \to 1$
- $(1,0) \to 1$

فرض کنیم یک پرسپترون تک لایه با وزن‌های w_1, w_2 و بایاس b وجود داشته باشد که بتواند این تابع را پیاده‌سازی کند. روابط زیر باید برقرار باشند:

۱. برای $(0,0)$: $0.w_1 + 0.w_2 + b < 0 \implies b < 0$

۲. برای $(1,1)$: $1.w_1 + 1.w_2 + b < 0 \implies w_1 + w_2 + b < 0$

۳. برای $(0,1)$: $0.w_1 + 1.w_2 + b \geq 0 \implies w_2 + b \geq 0$

۴. برای $(1,0)$: $1.w_1 + 0.w_2 + b \geq 0 \implies w_1 + b \geq 0$

اگر نامساوی‌های (۳) و (۴) را با هم جمع کنیم:

$$(w_2 + b) + (w_1 + b) \geq 0 \implies w_1 + w_2 + 2b \geq 0$$

از طرفی طبق نامساوی (۱)، b عددی منفی است. بنابراین اگر b را از عبارت فوق کم کنیم، سمت چپ بزرگتر می‌شود:

$$w_1 + w_2 + b > w_1 + w_2 + 2b \geq 0$$
$$\implies w_1 + w_2 + b > 0$$

این نتیجه با نامساوی (۲) که می‌گوید $w_1 + w_2 + b < 0$ در تناقض است. بنابراین هیچ خط راستی وجود ندارد که بتواند کلاس‌های خروجی ۰ و ۱ را در مسئله XOR از هم جدا کند (مسئله به صورت خطی تفکیک‌پذیر نیست). برای حل این مشکل حداقل به یک لایه مخفی (یعنی شبکه دو لایه) نیاز است.

د) معایب تابع پله‌ای و مزیت Sigmoid

چرا استفاده نمی‌شود؟ مشکل اصلی تابع فعال‌سازی پله‌ای (Step Function) در فرآیند آموزش شبکه است. الگوریتم‌های آموزش مدرن مانند «پس‌انتشار خطا» (Backpropagation) بر پایه محاسبه گرادیان (مشتق) تابع هزینه نسبت به وزن‌ها عمل می‌کنند. مشتق تابع پله‌ای در تمام نقاط (به جز صفر) برابر با صفر است و در نقطه صفر نیز تعریف نشده است (ضربه).

$$\frac{d\phi}{dx} = 0 \quad (\text{ for} x \neq 0)$$

وقتی مشتق صفر باشد، گرادیان صفر می‌شود و وزن‌ها به‌روزرسانی نمی‌شوند، در نتیجه شبکه چیزی یاد نمی‌گیرد.
مزیت تابع Sigmoid:

۱. **مشتق‌پذیری:** تابع سیگموئید $(\sigma(x) = \frac{1}{1+e^{-x}})$ در تمام دامنه خود مشتق‌پذیر و پیوسته است.

۲. **گرادیان غیرصفر:** مشتق آن همواره غیرصفر است (هرچند ممکن است کوچک شود)، که اجازه می‌دهد سیگنال خطا به لایه‌های عقب‌تر منتشر شود.

۳. **غیرخطی بودن نرم:** گذار نرم بین ۰ و ۱ ایجاد می‌کند که برای تقریب توابع پیچیده و احتمالات مناسب‌تر است.